SISTEM PENDETEKSI JENIS KENDARAAN PADA JALAN RAYA BERDASARKAN CITRA *WEB CAM* DENGAN MENGGUNAKAN METODE *SINGLE LAYER PERCEPTRON*

SKRIPSI

Diajukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh:
Andriana Agastyaningrum
NIM. 125150209111028

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2016

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

PENGESAHAN

SISTEM PENDETEKSI JENIS KENDARAAN PADA JALAN RAYA BERDASARKAN CITRA *WEB CAM* DENGAN MENGGUNAKAN METODE *SINGLE LAYER PERCEPTRON*

SKRIPSI

Diajukan untuk memenuhi persyaratan
Memperoleh gelar Sarjana Komputer

Disusun Oleh:
Andriana Agastyaningrum
NIM. 125150209111028

Skripsi ini telah diuji dan dinyatakan lulus pada
25 Agustus 2016
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I	Dosen Pembimbing II
Dr. Eng. Herman Tolle., S.T., M.T	Imam Cholissodin.,S.Si., M.Kom.
NIP. 19740823 200012 1 001	NIK. 201201 850719 1 001

Mengetahui,
Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan, S.T, M.T, Ph.D
NIP. 19710518 200312 1 001

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

PERNYATAAN ORISINALITAS

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

Apabila ternyata dalam naskah SKRIPSI ini dapat dibuktikan terdapat unsur-unsur PLAGIASI, saya bersedia SKRIPSI ini digugurkan dan gelar akademik yang saya peroleh (SARJANA) dibatalkan, serta diproses sesuai dengan perturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 09 Agustus 2016

Andriana Agastyaningrum
125150209111028

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Skripsi ini kutujukan kepada kedua orangtuaku

Ayahanda: Agus Suganda, S.Pd

Ibunda: Wagiatun

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

KATA PENGANTAR

Puji dan syukur penulis panjatkan kehadirat Tuhan yang Maha Esa, yang telah melimpahkan berkat dan anugerah-Nya kepada penulis sehingga penulis dapat menyelesaikan skripsi ini yang berjudul "**SISTEM PENDETEKSI JENIS KENDARAAN PADA JALAN RAYA BERDASARKAN CITRA *WEB CAM* DENGAN MENGGUNAKAN METODE *SINGLE LAYER PERCEPTRON***".

Pada penyusunan Skripsi ini tidak semata-mata hasil kerja penulis sendiri, melainkan juga berkat bimbingan dan dorongan dari pihak-pihak yang telah membantu, baik secara materi maupun non materi. Maka dari itu penulis ingin mengucapkan banyak terima kasih yang tak terhingga serta penghargaan yang setinggi-tingginya kepada orang-orang yang telah membantu penulis secara langsung maupun tidak langsung kepada yang terhormat:

1. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D selaku Ketua Teknik Jurusan Informatika.
2. Bapak Dr. Eng. Herman Tolle., S.T., M.T. selaku dosen pembimbing satu yang telah memberikan bimbingan dan dukungan serta arahan dalam menyelesaikan skripsi ini.
3. Bapak Imam Cholissodin, S.Si.,M.Kom selaku dosen pembimbing dua yang telah memberikan bimbingan dan dukungan serta rahan dalam menyelesaikan skripsi ini.
4. Bapak Kasyful Amron, T.T, M.Sc selaku dosen penasehat akademik penulis.
5. Ayahanda Agus Sugandi, S.Pd dan Ibunda tercinta Wagiatun yang telah memberikan doa, nasehat, kasih sayang dan kesabaranya dalam membesarkan dan mendidik penulis.
6. Eyang Sumijati yang selalu memberikan doa dan dukungannya dalam penyusunan skripsi ini.
7. Adinda Galuh Puspitaningrum yang selalu jadi penyemangat.
8. Fernanda Kharisma Editya, S.H yang selalu mendukung, membantu, memberikan semangat, dan meluangkan waktu serta nasehatnya setiap saat.
9. Vinda Arista Putri., S.Kom, Afe Faleniko., S.Kom, Galang Yuniar Mahendra., S.Kom yang telah membantu dalam penyelesaian skripsi dan bersedia memberikan waktunya.
10. Sahabatku Pratiwi Endah Permata, Meliza Latuputty, Yan Isnan, M.Nabaulhaq, Ilvi Nurdiana, Dwi Cahya Astriya Nugraha terimakasih semangat yang kalian tularkan disetiap saat.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

11. Teman – teman Teknik Informatika / Ilmu Komputer SAP angkatan 2012 yang telah memberikan masukan dan inspirasi kepada penulis selama menempuh studi dan menyelesaikan skripsi ini.
12. Teman – teman Laboratorium Komputasi Cerdas dan Visualisasi yang selalu memberi informasi untuk kelancaran skripsi ini.
13. Seluruh pihak yang telah membantu baik secara langsung maupun tidak, sekali lagi saya ucapkan terimakasih.

Akhir kata semoga skripsi ini dapat bermanfaat bagi kita semua dan menjadi bahan dalam pengembangan ilmu pengetahuan sebagai makna regenerasi bangsa untuk mencapai suatu keberhasilan khususnya untuk penulis sendiri.

Malang, 23 Agustus 2016

Penulis

nand.tya91@gmail.com

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

ABSTRAK

Jumlah alat transportasi yang semakin bertambah, menyebabkan kemacetan di beberapa titik terutama di persimpangan jalan. Hal ini diperparah dengan tidak adanya *timer traffic light* yang tidak sesuai dengan kepadatan dan tidak dibedakannya jalur khusus yang bisa dilalui roda dua, empat, atau lebih. Oleh karena itu, diperlukan sebuah sistem pengenalan bentuk kendaraan roda empat maupun roda dua yang dapat digunakan pada *traffic light*. Dalam kasus ini digunakan metode single layer perceptron untuk melakukan deteksi kendaraan roda empat maupun roda dua. Metode ini membutuhkan data *training* sebagai acuan untuk melakukan proses deteksi pada citra dengan merubah citra RGB menjadi hitam putih untuk dijadikan biner kemudian dilakukan proses perhitungan berdasarkan metode *Single Layer Perceptron*. Dari hasil pengujian, aplikasi mampu mendeteksi kendaraan roda empat dengan ketepatan prosentase 80% dan roda dua 88%. Sistem melakukan deteksi dengan *input* berupa *video* yang sudah diekstrak menjadi beberapa *citra* dan menghilangkan *background* pada *citra*.

Kata Kunci: *Jaringan Syaraf Tiruan, Single Layer Perceptron, Visual C# 2010*

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

repository.ub.ac.id

ABSTRACT

The increasing of transportation are causing congestion at some points especially at crossroad. This is compounded by the absence of traffic light timers that do not correspond to the traffic density and no different lines for two-wheel drivers, four, or more. Therefore, it needs a system for recognising four-wheel and two-wheel vehicles that can be used on a traffic light. In this case, the single layer perceptron method is used to detect the four-wheel and two-wheel vehicles. This method requires the training data as a reference for detection process in the image by changing the image RGB to black and white to be binary and then calculated based on the method of Single Layer Perceptron. In results, the application was able to detect a four-wheel vehicle with 80% accuracy and a two-wheel 88%. System detects the input in the form of video that has been extracted into some image and eliminates the images background.

Keywords: *Artificial Neural Networks, Single Layer Perceptron, Visual C# 2010*

UNIVERSITAS BRAWIJAYA

DAFTAR ISI

PENGESAHAN ii
PERNYATAAN ORISINALITAS iii
KATA PENGANTAR v
ABSTRAK vii
ABSTRACT viii
DAFTAR ISI ix
DAFTAR TABEL xi
DAFTAR GAMBAR xii
BAB I PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Rumusan Masalah 2
1.3 Batasan Masalah 2
1.4 Tujuan 2
1.5 Manfaat 2
1.6 Sistematika Penulisan 3
BAB II TINJAUAN PUSTAKA 4
2.1 Kajian Pustaka 4
2.2 Citra Digital 5
2.2.1 Tipe Citra Digital 6
2.3 Pengolahan Citra Digital 8
2.3.1 Operasi Pengolahan Citra 9
2.4 Pola 9
2.5 Pengenalan Pola 10
2.5.1 Jenis Kendaraan 11
2.6 Jaringan Syaraf Tiruan 11
2.6.1 Konsep Dasar Jaringan Syaraf Tiruan 12
2.6.2 Karakteristik Jaringan Saraf Tiruan 12
2.6.3 Kelebihan Jaringan Saraf Tiruan 13
2.6.4 Kelemahan Jaringan Syaraf Tiruan 13
2.7 Single Layer Perceptron 13
2.7.1 Arsitektur *Single Layer Perceptron* 14
2.7.2 Algoritma Pelatihan *Single Layer Perceptron* 15

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

2.7.3 Learning Rate 16
2.8 Evaluasi 16
BAB III METODOLOGI PENELITIAN 17
3.1 Studi Literatur 18
3.2 Pengambilan Data 18
3.3 Analisa Kebutuhan dan Perancangan Sistem 18
3.3.1 Analisa Kebutuhan 18
3.3.2 Perancangan Sistem 18
3.4 Pengujian 21
BAB IV PERANCANGAN SISTEM 22
4.1 Perancangan Sistem 22
4.1.1 Ubah *RGB* Ke *Biner* 23
4.1.2 Data Latih 24
4.1.3 *Single Layer Perceptron* 25
4.1.4 Uji Citra 26
4.2 Perhitungan Manual 27
4.3 Desain Interface 42
BAB V IMPLEMENTASI DAN PENGUJIAN 43
5.1 Implementasi Sistem 43
5.1.1 Spesifikasi Perangkat Keras 43
5.1.2 Spesifikasi Perangkat Lunak 43
5.2 Source Code Aplikasi 43
5.2.1 Source Code Dan Screenshot Extract Video 44
5.2.2 Source Code Dan Screenshot Pilih Citra 45
5.2.3 Source Code Dan Screenshot Data Latih 46
5.3 Pengujian Sistem 53
5.3.1 Pengujian Kendaraan Roda Empat 53
5.3.2 Pengujian Kendaraan Roda Dua 54
BAB VI PENUTUP 56
6.1 Kesimpulan 56
6.2 Saran 56
DAFTAR PUSTAKA 57
LAMPIRAN A PERHITUNGAN MANUAL 58
LAMPIRAN B DATA UJI KENDARAAN RODA EMPAT 135

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

DAFTAR TABEL

Tabel 4.1 Nilai Bobot, *Bias*, Target, *Learning Rate* 27
Tabel 4.2 Nilai *Biner* 28
Tabel 4.3 Nilai *Net* dan *FNet* Data Latih 1 30
Tabel 4.4 Perhitungan Bobot dan *Bias* Data Latih 1 31
Tabel 4.5 Nilai *Net* dan *FNet Epoch* ke-2 32
Tabel 4.6 Bobot dan *Bias* Baru *Epoch* ke-2 33
Tabel 4.7 Nilai *Net* dan *FNet* Data Latih 2 35
Tabel 4.8 Bobot dan *Bias* Data Latih 2 36
Tabel 4.9 *Net* dan *FNet Epoch* ke-2 37
Tabel 4.10 Bobot dan *Bias Epoch* ke-2 38
Tabel 4.11 Uji *Citra* Kendaraan Roda Dua 40
Tabel 5.1 Pengujian Akurasi Kendaraan Roda Empat 53
Tabel 5.2 *Confusion Matrix* 54
Tabel 5.3 Pengujian Akurasi Kendaraan Roda Dua 54
Tabel 5.4 *Confusion Matrix* 55

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

DAFTAR GAMBAR

Gambar 2.1 Citra Biner 7
Gambar 2.2 Citra *Grayscale* 7
Gambar 2.3 Citra Berwarna 8
Gambar 2.4 Tahapan Pengolahan Citra 8
Gambar 2.5 Pembatasan daerah positif dan daerah negatif 14
Gambar 2.6 Arsitektur Single Layer Perceptron 14
Gambar 3.1 Diagram Alir Tahapan Penelitian 17
Gambar 3.2 Diagram Alir Sistem 19
Gambar 4.1 *Flowchart* Sistem Secara Umum 22
Gambar 4.2 *Flowchart* Ubah *RGB* Ke *Biner* 23
Gambar 4.3 *Flowchart* Data Latih 24
Gambar 4.4 *Flowchart Single Layer Perceptron* 25
Gambar 4.5 *Flowchart* Uji Citra 26
Gambar 4.6 Diagram Alir Proses Iterasi 28
Gambar 4.7 Diagram Alir Proses Uji Citra 40
Gambar 4.8 Desain *Interface* Sistem Deteksi Kendaraan 42
Gambar 5.1 Struktur *Source Code Aplikasi* 44
Gambar 5.2 *Source Code* Pilih Citra 44
Gambar 5.3 *Screenshot* Extract *Video* Menjadi *Citra* 45
Gambar 5.4 *Source Code* Pilih Citra 45
Gambar 5.5 *Screenshot* Pilih Citra 46
Gambar 5.6 *Source Code* Pengubahan Gambar *RGB* menjadi Gambar *Biner* 46
Gambar 5.7 *Screenshot* Pengubahan Gambar *RGB* menjadi Gambar *Biner* 47
Gambar 5.8 *Source Code* Menampilkan Nilai *Biner* Dari Gambar 47
Gambar 5.9 *Screenshot* Tabel Data *Biner* X dan Y 48
Gambar 5.10 *Source Code* Perhitungan *Netx,Nety, Fnetx,* dan *Fnety* 48
Gambar 5.11 *Source Code Perhitungan Delta Bobot,Delta Bias, dan Perubahan Bobot* 49
Gambar 5.12 *Source Code* Menampilkan Hasil Perhitungan 50
Gambar 5.13 *Source Code* Menyimpan Data Bobot Ke Dalam *List Array* 50
Gambar 5. 14 *Screenshot* Perhitungan Data Latih Citra Kendaraan Mobil Dan Motor 51
Gambar 5.15 *Source Code* Uji Citra Deteksi Kendaraan 51

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

BAB I PENDAHULUAN

1.1 Latar Belakang

Jumlah alat transportasi semakin bertambah yang tidak diimbangi dengan bertambahnya pula panjang dan lebar jalan setiap tahunnya. Hal tersebut dikarenakan penambahan jumlah penduduk setiap harinya dimana setiap oang tingkat konsumtif terhadap kendaraan terus bertambah. Hal tersebut mengakibatkan penumpukan jumlah kendaraan dan terjadi kemacetan di titik-titik tertentu terutama di persimpangan jalan. Kondisi ini diperparah dengan adanya *timer traffic light* yang tidak sesuai dengan jumlah kendaraan yang melintasi persimpangan tersebut.

Pengaturan teknologi *traffic light* di Indonesia saat ini masih terus dikembangkan, sehingga tidak hanya berperan untuk mengurai kemacetan tetapi juga untuk meningkatkan keselamatan dalam berlalu lintas. Namun teknologi pada *traffic light* yang diterapkan saat ini masih belum optimal dalam mengatasi kemacetan lalu lintas, dikarenakan masih belum adanya sistem deteksi jenis kendaraan yang menyebabkan terjadinya kemacetan. Untuk pengoptimalan tersebut diperlukan sebuah sistem yang dapat digunakan pada *traffic light* untuk mengenali bentuk kendaraan roda empat maupun roda dua. Dengan adanya sistem pengenalan bentuk kendaraan ini, merupakan salah satu solusi yang baik untuk dikembangkan karena dapat mempermudah untuk mengetahui jenis kendaraan sehingga dapat ditentukan jalur mana yang boleh dilalui roda dua, empat atau lebih.

Untuk mengetahui jenis kendaraan diperlukan sebuah alat pendeteksi yang mana dapat berfungsi menangkap gambar dan memilah kendaraan tersebut dalam kategori apa. Pada penelitian sebelumnya tentang deteksi kendaraan untuk *vehicle counting* menggunakan *blob* analisis. Pada metode *blob* analisis setelah proses segmentasi dilakukan, kemudian dilakukan *update background.* Pada *feature extraction* digunakan analisis *bounding box* untuk mengenali objek yang bergerak, yang kemudian dilakukan *tracking*. Hasil penelitian ini mencapai akurasi 90% (Thou-Ho et.Al, 2006).

Pada penelitian sebelumnya pendeteksian yang dilakukan merupakan sebuah penelitian berbasis citra yang mana menggunakan konsep Pengenalan Citra Objek Sederhana Dengan Jaringan Saraf Tiruan Metode Perceptron. Penelitian tersebut mengenali objek sederhana seperti bentuk bangun ruang yang terdiri dari kubus, kerucut, tabung, prisma, dan limas dengan berbagai jenisnya. *Input* pada sistem ini berupa gambar dengan format *bitmap* yang di proses dari format RGB menjadi *grayscale*, yang kemudian semua perhitungan dan proses *interface* diolah melalui *software Borland Delphi* 7.0. Dari hasil pelatihan dan pengujian jaringan syaraf tiruan perceptron dapat mengenali objek dengan presentase tertinggi 92,65% (Ardi Pujiyanta, 2009).

Berdasarkan latar belakang di atas, metode *perceptron* dinilai cukup baik untuk dapat mengenali citra objek. Namun perlu pengembangan agar aplikasi tidak hanya mengenali pola melainkan melakukan deteksi objek melalui *citra*

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

web cam yang telah diekstrak. Hal ini dilakukan agar metode yang dikembangkan dalam pengolahan citra untuk proses deteksi dapat berjalan dengan maksimal. Metode *perceptron* ini memerlukan data *training* sebagai acuan untuk melakukan proses deteksi pada citra dengan mengubah *citra RGB* menjadi hitam putih untuk dijadikan biner, kemudian nilai biner dari citra kendaraan akan diproses berdasarkan metode *Single Layer Perceptron*. Sehingga pada penelitian ini dirancanglah suatu sistem yang dapat mendeteksi jenis kendaraan yaitu Sistem Pendeteksi Jenis Kendaraan Pada Jalan Raya Berdasarkan Citra *Web Cam* Menggunakan Metode *Single Layer Perceptron*.

1.2 Rumusan Masalah

Berdasarkan permasalahan yang dijelaskan pada bagian latar belakang, maka didapatkan rumusan masalah sebagai berikut:

1. Bagaimana rancangan dan implementasi sistem pendeteksi jenis kendaraan pada jalan raya berdasarkan citra *web cam* menggunakan metode *single layer perceptron*?
2. Bagaimana tingkat akurasi sistem pendeteksi jenis kendaraan pada jalan raya berdasarkan citra *web cam* menggunakan metode *single layer perceptron*?

1.3 Batasan Masalah

Dalam penelitian ini ada beberapa batasan masalah sebagai berikut.

1. Data dalam pengujian sistem ini yaitu file citra dengan format *bitmap* dan *png*.
2. Data diambil dari rekaman video kendaraan bukan *realtime* yang di ekstrak.
3. Citra diperoleh dengan menggunakan *Camera Web Cam type* Zavio B7210 2 *Megapixel Bullet IP Camera Progressive Scan CMOS Sensor*.
4. Range pengambilan video dilakukan pada pagi dan siang hari.
5. Sistem ini menangkap objek kendaraan roda empat dan roda dua.
6. Sistem hanya mengenali satu objek kendaraan, jika terdapat 2 objek dalam citra dan objek yang dikenali adalah objek yang pertama terdeteksi.
7. Objek hanya dapat dideteksi dengan pengambilan gambar dari depan.

1.4 Tujuan

Tujuan yang ingin diperoleh dari penelitian skripsi ini adalah:

1. Merancang dan mengimplemantasi sistem pendeteksi jenis kendaraan pada jalan raya berdasarkan citra *web cam* menggunakan metode *single layer perceptron*
2. Mengukur tingkat akurasi hasil deteksi jenis kendaraan dengan metode *single layer perceptron*.

1.5 Manfaat

Adapun manfaat yang diperoleh dari penelitian skripsi ini antara lain sebagai berikut:

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

1. Penulis dapat lebih mengetahui cara menerapkan ilmu-ilmu yang dipelajari dalam membuat sistem deteksi kendaraan dengan metode yang digunakan.
2. Pengguna mendapat wawasan mengenai pengembangan pengolahan citra.
3. Pengguna mendapat wawasan tentang penerapan pengolahan citra untuk aplikasi deteksi kendaraan roda empat dan bukan roda empat.

1.6 Sistematika Penulisan

Skripsi ini disusun berdasarkan sistematika penulisan sebagai berikut:

1. BAB I PENDAHULUAN
   Pada bab ini berisi latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, dan sistematika penulisan.
2. BAB II TINJAUAN PUSTAKA
   Pada bab ini dibahas mengenai kajian pustaka yang digunakan dalam pengerjaan skripsi. Adapun teori-teori yang terdapat dalam bab ini mencakup sistem, pengolahan citra, jaringan syaraf tiruan, *Single Layer Perceptron*, dan evaluasi.
3. BAB III METODOLOGI PENELITIAN
   Pada bab ini dibahas tentang metode-metode dan urutan langkah kerja yang digunakan untuk deteksi jenis kendaraan. Dalam metode penelitian terdiri dari studi literatur, analisis dan perancangan sistem, implementasi sistem, pengujian dan analisis serta penarikan kesimpulan.
4. BAB IV PERANCANGAN SISTEM
   Pada bab ini berisi perancangan sistem yang terdiri dari rangkaian proses kerja serta desain sistem menggunakan metode *Single Layer Perceptron.*
5. BAB V IMPLEMENTASI DAN PENGUJIAN
   Pada bab ini berisi tentang implementasi sistem pendeteksi kendaraan dengan citra *web cam* menggunakan *Single Layer Perceptron* dan pengujian dari implementasi metode yang digunakan.
6. BAB VI PENUTUP
   Pada bab ini berisi tentang kesimpulan berdasarkan seluruh rangkaian pengujian dan analisis yang dilakukan dalam penelitian, serta saran yang dapat digunakan untuk pengembangan sistem lebih lanjut.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

BAB II TINJAUAN PUSTAKA

Bab ini berisi tentang kajian pustaka dan dasar teori yang digunakan dalam skripsi mengenai implementasi metode *Perceptron* pada jaringan syaraf tiruan untuk mendeteksi kendaraan roda empat dan roda dua pada video lau lintas. Kajian pustaka membahas mengenai penelitian sekarang dan penelitian sebelumnya sehingga dapat saling berkaitan hasilnya. Dasar teori membahas tentang kebutuhan materi dalam penelitian yang disusun berdasarkan latar belakang dan rumusan terkait diantaranya adalah pengolahan citra digital, pola, pengenalan pola , jaringan syaraf tiruan, *single layer perceptron*, dan evaluasi.

2.1 Kajian Pustaka

Pada kajian pustaka ini membahas mengenai penelitian sebelumnya yang sesuai dengan skripsi ini. Diantaranya pada penelitian sebelumnya adalah pengenalan proses training menggunakan model jaringan syaraf tiruan dengan metode single layer perceptron yang merupakan metode pembelajaran dan pelatihan yang terdiri atas beberapa unit input yang hanya memiliki kemungkinan nilai -1, 0, dan 1. Metode perceptron secara umum adalah salah satu model dalam jaringan syaraf tiruan dengan lapisan tunggal (*single layer*). Metode tersebut nantinya dapat mengklasifikasikan suatu tipe pola yang sering disebut linear dan memiliki nilai bobot yang bias diatur dengan suatu nilai ambang sehingga akan terjadi pembatasan daerah positif dan daerah negative. Arsitektur jaringan syaraf tiruan single layer perceptron, yang terdiri dari lapisan input, bobot, maupun lapisan output, sehingga dapat melakukan generalisasi dari vector-vektor latihannya yang telah diinputkan. Hasil dari generalisasi tersebut dapat membuat sebuah nilai akurasi yang merupakan hasil penentuan baik atau buruk dari sebuah data (Erfanti, 2012).

Disisi lain metode perceptron juga dapat mengenali objek citra bentuk bangun ruang seperti kubus, kerucut, prisma dan limas yang memiliki berbagai jenisnya. Dimana pada penelitian yang lain ada yang melakukan sebuah pengujian menggunakan pasangan *input* berupa data *sample* sebanyak 25 buah dengan 20 gambar yang sudah dilatih dan 5 gambar yang belum dilatih sebelumnya. Hasil pengujian mendapatkan bahwa sistem dapat mengenali citra dari objek sederhana yang sudah dilatih sebelumnya dengan prosentase akurasi nilai tertinggi 92.65% dan prosentase terendah 60.75%. Sehingga dapat mengambil kesimpulan bahwa metode *perceptron* mampu mengenali objek dengan persentase ketepatan cukup baik (Pujiyanta, 2009).

Penelitian yang lain menggunakan metode blob analisis untuk *vehicle counting*, dimana proses pendeteksiannya dimulai dari segmentasi kemudian dilakukan *update background.* Pada *feature extraction* digunakan analisis *bounding box* untuk mengenali objek yang bergerak, yang kemudian dilakukan *tracking*. Selanjutnya pada proses *counting*, penelitian ini menggunakan *base line* (R1 dan R2) dimana setelah kendaraan melewati R2 akan ditangkap dan

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

kemudian dihitung. Hasil penelitian ini mencapai akurasi 90% (Thou-Ho et.all, 2006).

Selain itu ada penelitian lain yang juga berkaitan dengan skripsi ini yaitu menggunakan metode visi komputer untuk mendeteksi objek kendaraan dengan citra digital jalan raya, dimana proses pendeteksiannya merupakan gabungan dua metode *Hough Transform* (HT) dan *Connected Component Labeling* (CCL). HT menggunakan sebuah deteksi garis tepi pada jalan raya dengan dibantu oleh beberapa proses segmentasi dan deteksi tepi, sedangkan CCL menggabungkan beberapa proses yang lain seperti substraksi, *grayscalling*, pengambangan, *morfologi opening* dan *closing* sebelumnya sehingga dapat melakukan proses CCL. Pada proses pengujian mengambil 1 buah citra jalan kosong sebagai citra dasar dan 36 buah citra yang berisi kendaraan. Dari hasil pengujian diperoleh akurasi pendeteksian kendaraan berada pada prosentase 79.46%. Namun, untuk posisi tempat kendaraan yang berdekatan metode CCL masih mendeteksi dua kendaraan menjadi satu objek (Hadi, 2012).

Berdasarkan pada penelitian sebelumnya, didapatkan sebuah kajian untuk membuat sebuah sistem deteksi *kendaraan* roda empat dengan menggunakan metode *single layer perceptron*.

2.2 Citra Digital

Citra digital merupakan gambar dua dimensi yang menampilkan hasil dimonitor komputer sebagai himpunan berhingga (diskrit) nilai digital yang disebut pixel *(picture elements)*. Pixel merupakan hasil nilai intensitas warna yang di dapat dari sebuah elemen citra. Definisi dari citra digital adalah sebagai fungsi $f(x,y)$ dimana x dan y adalah koodinat spasial dan nilai $f(x,y)$ adalah intensitas citra pada koordinat tersebut.

Citra digital dapat dibedakan menjadi 2 jenis kelompok yaitu (Sutoyo, 2009):

1. Citra Bitmap *(bitmap image)*
   Citra digital yang dihasilkan oleh bentuk kumpulan pixel dalam array dua dimensi, seperti foto, gambar, lukisan dan hologram.
2. Grafik Vektor *(vector graphics)*
   Grafik Vektor merupakan salah satu jenis citra yang dibentuk menggunakan fungsi geometri dan matematika, seperti data berupa foto atau gambar yang ada di dalam file dan fungsi matematis yang mempresentasikan fungsi citra.

Citra Digital dapat didefinisikan dengan sebuah matriks berukuran $M \times N$ dimana perpotongan $M = kolom$ dan $N = baris$ disebut piksel yang memiliki 2 parameter yaitu intensitas dan koordinat. Nilai dalam koordinat (x,y) adalah $f(x,y)$, yaitu titik dari piksel besar intensitas atau warna. Maka dari itu, sebuah citra digital dapat ditulis dalam bentuk matriks sebagai berikut:

$$f(x,y)=\begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,M-1) \\ f(1,0) & \dots & \dots & f(1,M-1) \\ \dots & \dots & \dots & \dots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1,M-1) \end{bmatrix} \qquad 2.1$$

Pada proses digitalisasi (sampling dan kuantitas) diperoleh besar baris M dan kolom N sehingga hasil dari citra membentuk matriks $M \times N$ dan jumlah tingkat keabuan piksel G (Sutoyo, 2009).

Matriks merupakan sekumpulan bilangan yang tersusun di larik baris dan kolom. Pada umumnya penulisan notasi matriks menggunakan huruf kapital $A, B, \dots$. Jika matriks A terdiri dari m baris dan n kolom, maka ordo $M \times N$ dapat ditulis dengan persamaan (2.2):

$$A = a_{ij} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \qquad 2.2$$

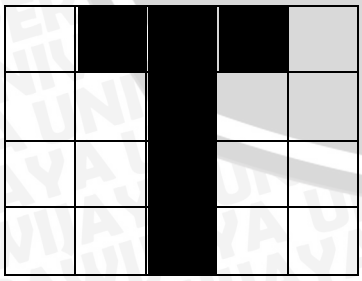
2.2.1 Tipe Citra Digital

Sesuai dengan format penyimpanan nilai warnanya, citra digital dibagi dalam 3 tipe, yaitu (Muhammad, 2009):

1. Citra Biner

   Citra biner merupakan citra yang memiliki hanya dua nilai yaitu 0 dan 1. Selain itu warna yang dimiliki oleh citra biner ada 2, dimana setiap piksel pada citra yang bernilai 0 untuk warna hitam dan bernilai 1 untuk warna putih. Pada citra biner membutuhkan 1 bit dalam penyimpanan kedua warna di memori. Berikut ini contoh gambar dalam citra biner dapat dilihat pada Gambar 2.1.

Contoh:



Citra Biner (hitam = 0, putih = 1)
= 1 0 0 0 1
= 1 1 0 1 1
= 1 1 0 1 1
= 1 1 0 1 1

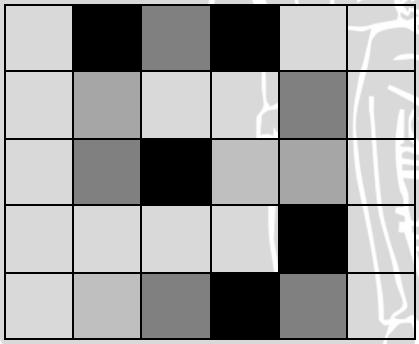
Gambar 2.1 Citra Biner

2. Citra *Grayscale*

Citra *grayscale* merupakan citra dengan tingkat derajat keabuan tertentu. Warna pada citra *grayscale* antara hitam dan putih, tetapi warnanya memiliki variasi yang sangat banyak. Warna-warna tersebut berdasarkan jumlah bit yang disediakan dalam suatu memori. Untuk citra yang menggunakan jumlah maksimum 8 bit maka kemungkinan warna sebanyak 2^8 = 256 dimana yang hitam pekat 0 dan yang terputih 256, dapat didefinisikan sebagai berikut:

$$f(x,y)\sum\{0...256\} \qquad 2.3$$

Contoh citra *grayscale* pada derajat keabuan 4 bit mewakili warna gradasi sebanyak $2^4 = 16$ dengan nilai 0 adalah warna hitam dan nilai 15 adalah warna hitam.



Skala Keabuan 4 bit (hitam=0, putih=15)

= 15 0 6 0 13 15
= 15 10 15 15 6 15
= 15 6 0 12 13 15
= 15 13 15 15 0 15
= 15 12 6 0 6 15

Gambar 2.2 Citra *Grayscale*

3. Citra Berwarna

Citra berwarna merupakan citra standart (Bitmap) dengan ukuran 512 × 512 dan pada kedalaman warna 24 bit (*true color*) dengan R=8 bit, G=8 bit, B=8 bit. Setiap warna dasar dengan penyimpanan 8 bit=1 *byte* mempunyai gradasi warna sebanyak 255 warna, sehingga pada setiap piksel memiliki kombinasi warna sebanyak $2^8.2^8.2^8 = 2^{24} = 16$ juta warna yang dinyatakan per piksel pada:

Bit ke-0 sampai ke-7 untuk warna merah (*Red*)
Bit ke-8 sampai ke-15 untuk warna hijau (*Green*)
Bit ke-16 sampai ke-23 untuk warna biru (*Blue*)

Banyak warna dasar merupakan susunan setiap piksel pada citra digital yang dinyatakan dengan persamaan berikut:

$$f(x,y)\sum\{0...256\}$$
$$f(x,y)\sum\{0...256\}$$
$$f(x,y)\sum\{0...256\} \qquad 2.4$$

Berikut ini contoh gambar citra berwarna dapat dilihat pada Gambar 2.3.

Gambar 2.3 Citra Berwarna

2.3 Pengolahan Citra Digital

Pengolahan citra digital adalah pengolahan dan analisis citra yang saling berkaitan dengan melibatkan persepsi visual. Secara umum, pengolahan citra digital dalam prose 2 dimensinya memiliki tujuan memperbaiki kualitas citra agar mudah direprentasikan oleh manusia maupun mesin. Proses ini mempunyai ciri data masukan (*input*) dan informasi keluaran (*output*) yang meghasilkan sebuah citra, dimana citra keluaran (*output*) mempunyai kualitas yang lebih baik dari pada citra masukan (*input*).

Gambar 2.4 Tahapan Pengolahan Citra

2.3.1 Operasi Pengolahan Citra

Secara umum, operasi pada pengolahan citra memiliki klasifikasi sebagai berikut:

1. Perbaikan kualitas citra (*image enhancement*)
   Operasi ini berguna untuk memperbaiki kualitas citra yang mendasar pada ciri-ciri khusus. Contoh operasi pada *image enhancement* yaitu:
   - Perbaikan kontras gelap atau terang
   - Perbaikan tepian objek (*edge enhancement*)
   - Penajaman (*sharpening*)
   - Pemberian warna semu (*pseudocoloring*)
   - Penapisan derau (*noise filtering*)
2. Pemugaran citra (*image restoration*)
   Operasi ini digunakan dalam menghilangkan atau meminimalkan kesalahan di sebuah citra. Contoh operasi pemugaran citra, yaitu penghilangan kesamaran (*deblurring*) dan penghilangan derau (*noise*).
3. Pemampatan citra (*image compression*)
   Proses *image compression* citra yang digunakan harus tetap memiliki kualitas gambar lebih bagus. Contohnya ketika suatu file .BMP (*bitmap*) ukuran 258 KB diproses dengan metode JPEG menghasilkan ukuran 49 KB.
4. Segmentasi citra (*image segmentation*)
   Proses ini digunakan memecahkan suatu citra ke dalam beberapa segmen dengan suatu kriteria tertentu seperti letak piksel dan intensitas yang masih berdekatan. Algoritma yang digunakan dalam proses segmentasi citra, diantaranya algoritma *Region Growing*, *JSEG*, *Watershead*, *K-Mean Clustering*, *Connected Component Labeling* (CCL) dan lain sebagainya.
5. Pengorakan citra (*image analysis*)
   Operasi pengorakan citra melakukan ekstrasi untuk membantu identifikasi objek. Proses ini diperlukan untuk mendekatkan objek yang diinginkan dengan lingkungan sekelilingnya. Contoh operasi pengorakan citra, yaitu :
   - Pendeteksian tepi objek (*edge detection*)
   - Ekstraksi batas (*boundary*)
   - Representasi daerah (*region*)
6. Rekonstruksi citra (*image reconstruction*)
   Operasi ini digunakan dalam menggabungkan citra yang satu dengan citra yang lain. Pemanfaatan operasi ini dalam bidang medis, dapat dilihat pada foto *rontgen* yang menggunakan hasilnya untuk membentuk ulang gambar organ tubuh.

2.4 Pola

Pola adalah entitas yang didefinisikan dan diidentifikasikan dan diberi nama melalui ciri-cirinya. Ciri-ciri tersebut dipakai sebagai unsur pembeda satu pola dengan pola lainnya. Ciri yang memiliki daya pembeda yang tinggi adalah ciri yang baik, sehingga dalam mengelompokkan pola lebih mudah dengan melihat ciri yang dimiliki dari hasil akurat yang tinggi. (Hanif, 2009).

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Ciri-ciri pada pola didapat melalui hasil pengukuran pada titik objek uji, dimana ciri-ciri tersebut dapat diperoleh dari beberapa informasi diantaranya:

a. Spasial, seperti intensitas pixel dan histogram
b. Tepi, seperti kekuatan dan arah
c. Kontur, seperti garis, elips dan lingkaran
d. Wilayah atau bentuk, seperti keliling, luas dan pusat massa
e. Hasil tranformasi fourier, frekuensi

Pola merupakan komposit atau gabungan dari ciri yang berupa sifat dari sebuah objek. Beberapa contoh pola, yaitu (Hanif, 2009):

a. Huruf, mempunyai ciri-ciri lengkungan garis, tinggi, titik sudut, dan tebal.
b. Suara, mempunyai ciri-ciri intonasi, frekuensi nada, dan amplitudo.
c. Tanda tangan, mempunyai ciri-ciri tekanan, panjang, dan kerumitan.
d. Sidik jari, mempunyai ciri-ciri seperti jumlah garis, dan lengkungan.

2.5 Pengenalan Pola

Pengenalan pola memiliki tujuan untuk menentukan kategori atau kelompok pola didasarkan pada ciri-ciri pola tersebut. Pengenalan pola dapat juga membedakan objek satu dengan objek yang lainnya.

Pengenalan pola itu sendiri adalah cabang dari kecerdasan buatan (*Artificial Inteligence*). Di bawah ini ada beberapa definisi dari pengenalan pola, di antaranya:

a. Suatu ilmu untuk mengklasifikasikan atau menggambarkan sesuatu berdasarkan pengukuran kuantitatif fitur (ciri) atau sifat utama dari suatu obyek. (Darma, 2010).
b. Penentuan suatu kejadian atau objek fisik ke dalam salah satu atau beberapa kategori. (Hanif, 2009).

Berdasarkan pengertian tersebut, pengenalan pola juga dapat diartikan sebagai cabang kecerdasan buatan yang memfokuskan pada metode pengklasifikasian objek ke dalam kelas-kelas tertentu untuk menyelesaikan masalah tertentu. Saat ini aplikasi-aplikasi pengenalan pola sudah sangat bervariasi, di antaranya sebagai berikut (Darma, 2010):

a. *Voice recognition* yang menggunakan pengenalan suara sebagai kunci bagi pengguna sistem.
b. *Fingerprint identification* yang menggunakan pengenalan sidik jari sebagai kunci telah dipakai oleh masyarakat/instansi sebagai pengganti pin atau *password* untuk mengakses suatu sistem tertentu.
c. *Face identification* yang menggunakan pengenalan wajah bagi pengguna sistem, bahkan pada era modern sekarang ini badan penegak hukum sedang melakukan pengembangan sistem untuk mengidenfikasi para penjahat yang menjadi target operasi dengan menggunakan *scanning* pada wajah para pelaku yang sudah disimpan.
d. *Handwriting identification* menggunakan pengenalan tulisan yang telah banyak digunakan oleh sistem perbankan untuk membuktikan siapa saja yang berhak melakukan transaksi perbankan tersebut.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

e. *Optical Character Recognition (OCR)* yang sudah banyak digunakan oleh *counter* pengecekan barang.
f. *Robot vision* digunakan oleh aplikasi robotik untuk mengenali objek tertentu pada suatu lingkungan atau daerah yang unik.

2.5.1 Jenis Kendaraan

Jenis kendaraan bermotor, yaitu (UU RI No. 22 Tahun 2009) :

a. Sepeda motor adalah kendaraan bermotor beroda dua dengan atau tanpa rumah-rumah dan dengan atau tanpa kereta samping atau kendaraan bermotor beroda tiga tanpa rumah-rumah.
b. Mobil penumpang adalah setiap kendaraan bermotor yang dilengkapi sebanyak-banyaknya 8 (delapan) tempat duduk tidak termasuk tempat duduk pengemudi, baik dengan maupun tanpa perlengkapan pengangkutan bagasi.
c. Mobil bus adalah setiap kendaraan bermotor yang dilengkapi lebih dari 8 (delapan) tempat duduk tidak termasuk tempat duduk pengemudi, baik dengan maupun tanpa perlengkapan pengangkutan bagasi.
d. Mobil barang adalah setiap kendaraan bermotor selain dari yang termasuk dalam sepeda motor, mobil penumpang dan mobil bus.
e. Kendaraan khusus adalah kendaraan bermotor selain daripada kendaraan bermotor untuk penumpang dan kendaraan bermotor untuk barang, yang penggunaannya untuk keperluan khusus atau mengangkut barang-barang khusus.

2.6 Jaringan Syaraf Tiruan

Jaringan syaraf tiruan (*artifical neural network*) adalah sistem komputasi yang arsitektur dan pengoperasinya didasarkan oleh pengetahuan tentang sel syaraf biologis yang terdapat di otak. Jaringan syaraf tiruan merupakan salah satu representasi buatan dari otak manusia yang selalu mencoba menstimulasi proses belajar pada otak manusia (Samudera, 2005).

Jaringan syaraf tiruan (JST) merupakan sistem yang memproses informasi dengan mengadopsi cara kerja otak manusia (Fausset, 1994). Jaringan syaraf tiruan cenderung menyimpan pengetahuan yang memiliki sifat pengalaman dan membuatnya siap ketika akan digunakan. Ada tiga elemen penting dalam JST yaitu (Rojas, 1996):

a. Pola hubungan antar neuron beserta Arsitektur Jaringan.
b. Menemukan bobot jaringan menggunakan Algoritma Pembelajaran.
c. Penggunaan fungsi aktivasi.

JST sebagian besar terdiri dari elemen pemrosesan sederhana yang biasa disebut *neurons, cells* atau *node.* Proses pengolahan informasi dalam JST terjadi pada neuron-neuron. Sinyal antara neuron-neuron diteruskan melalui *link-link* yang terhubung dan memiliki bobot terisolasi. Kemudian setiap neuron menerapkan fungsi aktivasi terhadap input jaringan.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

JST memiliki struktur tersebar paralel yang sangat besar dan memiliki kemampuan belajar, hingga dapat melakukan generalisasi, sehingga dapat menghasilkan output yang benar untuk input yang belum pernah dilatihkan. Jadi JST mampu menyelesaikan masalah-masalah yang terlalu kompleks (Suyanto, 2014).

Model jaringan syaraf dibuktikan dengan kemampuan untuk mengemulasi, menganalisis, memprediksi dan mengasosiasi. Kemampuan yang ada di dalam jaringan syaraf tiruan dapat dipakai untuk belajar serta menghasilkan aturan atau *input*, sehingga membuat prediksi dari kemungkinan *output* yang akan muncul (Hermawan, 2006).

2.6.1 Konsep Dasar Jaringan Syaraf Tiruan

Jaringan syaraf tiruan terdiri dari beberapa *neuron* yang saling berhubungan seperti otak manusia. *Neuron* atau sel syaraf merupakan sebuah unit dasar operasi jaringan syaraf tiruan. Hubungan antar *neuron* disebut dengan nama bobot. *Neuron* akan merubah informasi yang masuk dan yang mengarah ke *neuron* lainnya. Informasi itu disimpan dalam suatu nilai tertentu dalam bobot yang dimaksud (Kusumadewi, 2003).

2.6.2 Karakteristik Jaringan Saraf Tiruan

Pemrograman tidak diperlukan dalam penyelesaian masalah yang menggunakan jaringan syaraf tiruan. Penyelesaian masalah jaringan syaraf tiruan dapat melalui proses mempelajari beberapa pelatihan yang didapat. Biasanya jaringan saraf tiruan diberikan sebuah himpunan pola pelatihan yang terdiri dari beberapa kumpulan contoh pola. Rangkaian contoh pola yang didapat merupakan awal dari proses pembelajaran jaringan syaraf tiruan. Adapun karakteristik dari jaringan syaraf tiruan adalah:

a. Faktor Bobot

   Bobot yaitu nilai yang mendeskripsikan kepentingan atau tingkat hubungan antara satu *node* dengan *node* lainnya. Jika bobot semakin besar maka semakin penting pula hubungan antara *node* tersebut. Bobot adalah suatu hubungan berupa bilangan *integer* maupun *real*, tergantung dari model yang digunakan dan jenis dari masalah. Bobot-bobot tersebut bisa ditentukan untuk berada didalam interval tertentu. Bobot dapat menyesuaikan dengan pola *input* dalam proses pelatihan. Karena adanya kemampuan untuk belajar, maka jaringan dapat memperbaiki diri secara terus-menerus. Jika ada masalah baru, jaringan juga dapat beradaptasi dan belajar dari masalah itu dengan mengatur kembali nilai bobot agar sesuai dengan karakter nilai. (Puspaningrum, 2006).

b. Fungsi Aktivasi

   Tiap-tiap *neuron* memiliki situasi internal yang sering disebut level aktivitas atau aktivasi berupa fungsi input yang diterima. Secara tipe suatu *neuron* mengirim aktivitasnya ke *neuron-neuron* lain sebagai sinyal. *Neuron* hanya

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

dapat mengirimkan satu sinyal dan hanya sesaat, meskipun sinyal tersebut bisa dipancarkan ke *neuron-neuron* lain (Puspaningrum, 2006).

2.6.3 Kelebihan Jaringan Saraf Tiruan

Kelebihan jaringan saraf tiruan adalah sebagai berikut:

a. Mampu mengakuisi pengetahuan meskipun belum ada kepastian.
b. Mampu melakukan ekstraksi dan generalisasi dari suatu pola tertentu.
c. Pola pengetahuan dapat diciptakan JST melalui kemampuan belajar atau pengaturan diri.
d. Memiliki fault tolerance (toleransi kesalahan), gangguan dianggap sebagai *noise* saja.
e. Perhitungan dilakukan secara pararel yang menjadikan proses menjadi lebih singkat.

Selain itu JST mampu melakukan beberapa hal berikut ini:

a. Klasifikasi yaitu memilih input data ke dalam kategori yang ditentukan dan yang telah ditetapkan.
b. Asosiasi yaitu hanya dengan bagian dari obyek lain dapat menggambarkan suatu obyek secara keseluruhan.
c. *Self organizing* yaitu kemampuan memproses *input* tanpa harus memiliki target.
d. Optimasi yaitu meminimalisir fungsi biaya dengan menemukan suatu jawaban terbaik.

2.6.4 Kelemahan Jaringan Syaraf Tiruan

Terdapat pula kelemahan pada jaringan syaraf tiruan diantaranya adalah:

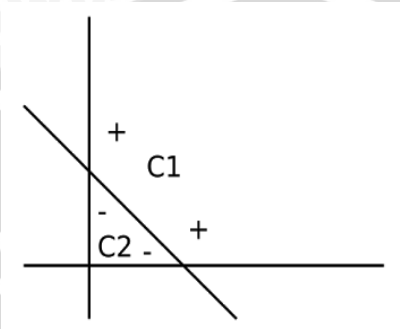
a. Kurang dapat untuk melakukan operasi-operasi numerik yang presisinya tinggi
b. Kurang dapat melakukan operasi algoritma aritmatik, operasi logika dan simbolis
c. Proses *training* membutuhkan waktu lama ketika terjadi jumlah data yang besar.

2.7 Single Layer Perceptron

Pada jaringan syaraf tiruan, terdapat beberapa arsitektur salah satunya adalah *single layer*. Jaringan *single layer* akan menerima sebuah *input* yang kemudian akan diolah secara langsung menjadi *output* tanpa melalui lapisan tersembunyi. *Perceptron* merupakan salah satu bentuk jaringan syaraf yang sederhana. *Perceptron* berfungsi untuk mengklasifikasi suatu tipe pola tertentu atau biasa dikenal dengan pemisahan secara linier. Jadi *single layer perceptron* bisa diartikan sebagai salah satu teknik jaringan syaraf tiruan yang hanya memiliki lapisan *input* dan lapisan output.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

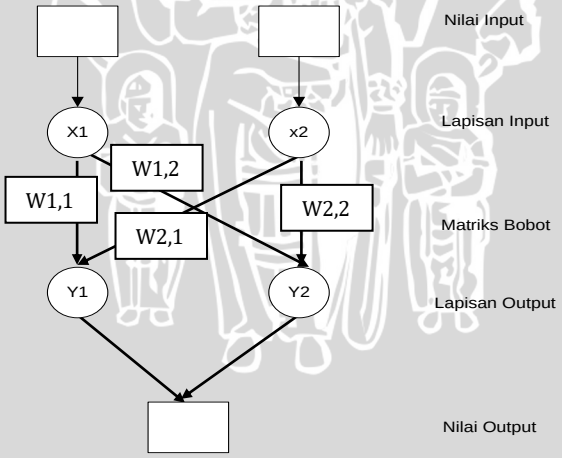
Jaringan *single layer perceptron* termasuk pembelajaran terawasi *(supervised learning)*, dimana untuk nilai *output* sudah diketahui sebelumnya. Pada dasarnya, jaringan ini mempunyai nilai bobot yang bisa diatur dan nilai ambang (*threshold*). Pada fungsi aktivasi nilai *threshold* adalah non negatif. Seperti yang terlihat pada Gambar 2.5, dimana terjadi pembatasan antara daerah negative dan positif (Kusumadewi, 2003).



Gambar 2.5 Pembatasan daerah positif dan daerah negatif

2.7.1 Arsitektur *Single Layer Perceptron*

Pada arsitektur *single layer perceptron* terdapat 2 layer yaitu *layer input* dan *layer output*. Pada Gambar 2.6 dijelaskan bagaimana jaringan *single layer* menerima *input* yang kemudian akan diolah menjadi *output* secara langsung.



Gambar 2.6 Arsitektur Single Layer Perceptron

Terdapat 3 lapisan *neuron* pada lapisan *input* yaitu X_1, X_2, dan X_3 sedangkan terdapat 2 *neuron* pada lapisan *output* yaitu Y_1 dan Y_2. *Neuron* pada lapisan *input* dan *output* saling berhubungan, dimana hubungan keduanya ditentukan oleh bobot yang bersesuaian (Kusumadewi, 2003).

2.7.2 Algoritma Pelatihan *Single Layer Perceptron*

Berikut adalah algoritma pelatihan *single layer perceptron* (Sri Kusuma, 2003):

a. Inisialisasi semua bobot dan bias (untuk sederhananya set 0), set *learning rate*

b. Selama kondisi berhenti bernilai *false*, maka lakukan langkah-langkah berikut:

1. Set input dengan nilai sama dengan vektor *input*
2. Hitung respon untuk *unit output*:

$$y_{in} = b + \sum \omega_i x_i \qquad 2.5$$

$$y = \begin{cases} 1 \ , y_{in} > 0 \\ 0 \ , y_{in} = 0 \\ -1, y_{in} < 0 \end{cases} \qquad 2.6$$

Dimana,

y_{in} = nilai net
y = nilai *fnet*
b = nilai bias awal
ω_i = bobot vektor
x_i = *input* data ke-*i*, *i* = 1,2,...,*n*

3. Perbaiki bobot dan bias jika terjadi *error*

Jika $y \neq t$, maka:

$$\omega_i(baru) = \omega_i(lama) + \alpha * t * x_i \qquad 2.7$$

$$b(baru) = b(lama) + \alpha * t \qquad 2.8$$

Jika tidak, maka:

$$\omega_i(baru) = \omega_i(lama) \qquad 2.9$$

$$b(baru) = b(lama) \qquad 2.10$$

Dimana,

ω_i = bobot vektor
b = nilai bias
α = learning rate
t = nilai target
x_i = *input* data ke-*i*, *i* = 1,2,...,*n*

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

c. Tes kondisi berhenti: kondisi berhenti *TRUE* jika tidak terjadi perubahan bobot pada (i), sebaliknya kondisi berhenti *FALSE* jika masih terjadi perubahan bobot.

Untuk *input biner* maupun *bipolar* dapat digunakan algoritma di atas dengan *threshold* tertentu, dan bias yang dapat diatur. Selanjutnya algoritma memperbaiki nilai bobot yang berhubungan dengan *input* yang aktif $x_i \neq 0$ dan nilai bobot yang tidak menghasilkan nilai y yang benar.

2.7.3 Learning Rate

Learning rate mempunyai pengaruh signifikan terhadap waktu yang dibutuhkan untuk mencapai target. Hal ini membuat nilai perubahan bobot menjadi optimal dan *error* yang dihasilkan menjadi lebih kecil. Variabel *learning rate* adalah $(0 < \alpha \leq 1)$ (Nugraha, 2004).

Apabila penggunaan nilai *learning rate* terlalu kecil maka epoh yang dibutuhkan terlalu banyak untuk mencapai nilai target yang diinginkan, sehingga proses *training* membutuhkan waktu lama. Jika nilai *learning rate yang dipakai* semakin besar maka semakin cepat pula proses pelatihannya. Namun jika terlampau besar maka akan berakibat berulangnya nilai *error* antara nilai tertentu dan jaringan tidak stabil.

2.8 Evaluasi

Evaluasi atau pengujian merupakan proses untuk mengukur tingkat keberhasilan atau membandingkan suatu sistem yang diimplementasikan dengan standart yang telah ditetapkan. Tujuan evaluasi percobaan pada klasifikasi yaitu untuk mengukur keefektifan apakah sistem mengklasifikasi secara benar (Prasetyo, 2014). Akurasi adalah nilai seberapa dekat atau seberapa miripnya nilai hasil pengukuran dengan nilai sebenarnya adapun persamaannya adalah pada Persamaan 2.11 (Selly, 2013).

$$Akurasi = \frac{\sum Data\ Testing\ Benar\ dokumen\ relevan\ terretrieve}{\sum Jumlah\ Total\ Data\ Testing} \times 100\% \qquad 2.11$$

BAB III METODOLOGI PENELITIAN

Bab ini menjelaskan tentang proses dari perangkat lunak sistem pendeteksi jenis kendaraan pada jalan raya berdasarkan citra *web cam* dengan menggunakan metode *single layer perceptron*. Terdapat beberapa tahap yang diterapkan dalam penelitian ini yang meliputi studi literatur, penyusunan data teori, analisa sistem, perancangan sistem, dan pengujian sistem.

Gambar 3.1 Diagram Alir Tahapan Penelitian

Penjelasan Gambar 3.1 Diagram Alir Tahapan Penelitian yaitu:

1. Mempelajari studi literatur yang berhubungan dengan metode yang digunakan, yaitu metode jaringan syaraf tiruan *single layer perceptron* untuk melakukan deteksi jenis kendaraan roda empat dan roda dua.
2. Melakukan pengumpulan data yang dibutuhkan dalam penunjang penulisan skripsi ini.
3. Melakukan analisis kebutuhan dan perancangan sistem yang berhubungan dengan pengembangan aplikasi yang akan dibuat.
4. Melakukan implementasi perangkat lunak berdasarkan analisis dan perancangan sistem yang telah dibuat.
5. Melakukan pengujian pada perangkat lunak pendeteksi jenis kendaraan menggunakan metode *single layer perceptron*.
6. Melakukan pengambilan keputusan atas sistem yang telah dibuat.

3.1 Studi Literatur

Studi literatur mempelajari tentang penjelasan dasar teori yang digunakan untuk menunjang dalam penulisan skripsi. Teori pendukung tersebut yaitu buku, jurnal, *e-book*, dan penelitian yang pernah dilakukan sebelumnya yang terkait dengan topik skripsi ini.

Dasar teori yang digunakan untuk menunjang dalam penyusunan penulisan skripsi ini adalah metode jaringan syaraf tiruan *single layer perceptron* yang akan diterapkan dalam perangkat lunak untuk melakukan deteksi jenis kendaraan roda empat dan roda dua.

3.2 Pengambilan Data

Data yang akan digunakan dan diolah perangkat lunak ini adalah data berupa citra dengan format *png* yang sudah disiapkan sebelumnya. Data gambar ini berisi kondisi lalu lintas kendaraan yang ada di jalan raya, yaitu objek kendaraan bermotor roda dua maupun roda empat.

3.3 Analisa Kebutuhan dan Perancangan Sistem

Pada bab ini menjelaskan tentang kebutuhan dan perancangan yang digunakan untuk membangun perangkat lunak deteksi jenis kendaraan.

3.3.1 Analisa Kebutuhan

Analisa kebutuhan membahas semua hal terkait untuk mengetahui apa saja yang diperlukan dalam membangun sistem. Pada tahap ini akan melakukan analisis terhadap semua spesifikasi kebutuhan yang diinginkan oleh pengguna. Hasil dari analisis kebutuhan tersebut berupa informasi yang akan digunakan dalam membangun sebuah sistem agar sesuai dengan harapan pengguna.

Berikut ini adalah analisis kebutuhan dalam mengimplementasikan perangkat lunak yang akan dibangun. Perangkat lunak yang dibangun memiliki beberapa fungsi sebagai berikut:

1. Perangkat lunak yang dikembangkan harus memiliki tampilan yang dapat memudahkan pengguna dalam menggunakan perangkat lunak tersebut.
2. Perangkat lunak dapat melakukan proses deteksi jenis kendaraan secara otomatis ketika citra diubah ke *biner* dan diproses dengan metode *single layer perceptron*.

3.3.2 Perancangan Sistem

Pada bagian perancangan ini menjelaskan tentang tahap yang dilakukan untuk membentuk sistem agar sesuai dengan harapan. Sistem ini akan dibangun dengan menggunakan salah satu metode jaringan syaraf tiruan, yaitu *single layer perceptron*.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

3.3.2.1 Deskripsi Umum Perangkat Lunak

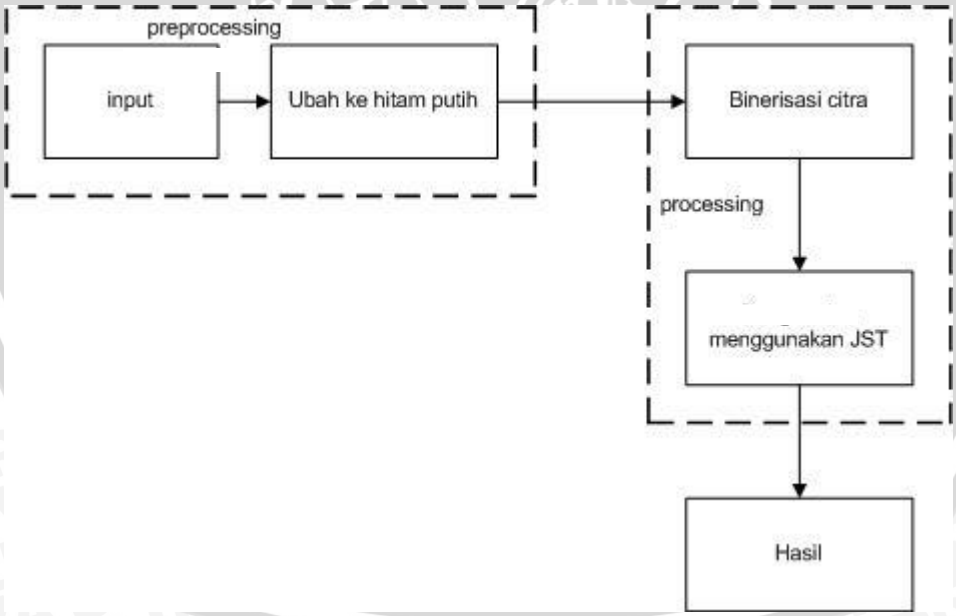
Penelitian ini dilakukan dengan menerapkan salah satu metode jaringan syaraf tiruan, yaitu *single layer* perceptron yang bertujuan untuk dapat melakukan proses deteksi jenis kendaraan berdasarkan citra *web cam* dengan menggunakan metode tersebut.

Pengguna memasukkan citra kendaraan dengan format *png* yang kemudian diubah menjadi *biner*, setelah itu dilakukan proses perhitungan menggunakan metode *single layer perceptron* untuk mengetahui jenis kendaraan yang terdeteksi oleh sistem.

3.3.2.2 Perancangan Perangkat Lunak

Pada perancangan perangkat lunak, *compiler* yang digunakan adalah *Microsoft Visual C# 2010.* Dalam pembuatan program, terdapat bagian-bagian penting dalam setiap langkah sesuai dengan algoritma mulai dari awal sampai *output*.

Pada prosesnya terdapat beberapa algoritma program secara global yang terdiri dari 5 proses yaitu proses input, mengubah ke hitam putih, *binerisasi* citra, metode jaringan syaraf tiruan *single layer perceptron*, dan hasil seperti yang ditunjukkan pada Gambar 3.2.



Gambar 3.2 Diagram Alir Sistem

Proses dalam pembuatan sistem yang dilakukan seperti terlihat pada Gambar 3.2 yang terdiri dari 3 tahap, yaitu:

a. Preprocessing

Pada tahap *preprocessing* meliputi proses *input* dan proses ubah ke hitam putih seperti berikut:

1. *Input*
   Input merupakan proses awal pada tahap *preprocessing*. Tahap ini akan melakukan *input* data yang akan dilatih berupa citra jenis kendaraan. Citra jenis kendaraan pada saat diinputkan masih berupa citra *RGB*. Dimana *RGB* adalah warna dasar yang dapat diterima oleh mata manusia, yang merupakan kombinasi dari ketiga warna dasar *RGB* pada setiap piksel. Intensitas nilai minimum pada warna dasar nol (0) dan nilai maksimum 255 (8 bit).
2. Ubah ke Hitam Putih
   Proses selanjutnya pada tahap *preprocessing* setelah dilakukan *input* citra kendaraan yaitu mengubah citra yang masih berupa *RGB* menjadi citra hitam putih dan memberi nilai biner 0 menjadi 1 sedangkan nilai 1 menjadi -1 berdasarkan target roda dua dan roda empat.

b. Processing
   1. Binerisasi citra
      Proses binerisasi citra akan mengubah citra ke dalam bentuk biner (0 dan 1). Proses ini dilakukan dengan membulatkan setiap nilai keabuan dari piksel yang berada di atas atau di bawah nilai ambang. Untuk menentukan besarnya nilai ambang dilakukan metode *thresholding,* sehingga derajat keabuan bisa diubah sesuai kebutuhan. Selanjutnya nilai hasil pembineran akan digunakan sebagi *input* pada proses selanjutnya.
   2. Pengenalan Melalui JST
      Pada proses pengenalan melalui jaringan saraf tiruan menggunakan metode *single layer perceptron*, ada tiga tahap yang dilakukan yaitu :
      - Tahap pertama: Pemberian *input* untuk proses *learning* dimana *input* berupa nilai *biner* yang sudah didapat dari proses binerisasi citra sebelumnya.
      - Tahap kedua: Mempelajari *error,* setelah inputan diproses melalui perhitungan dengan metode *single layer perceptron* akan didapatkan hasil dimana hasil tersebut akan dicocokan dengan target yang sudah ditentukan. Apabila hasil perhitungan sama dengan target yang sudah ditentukan maka bobot akan nol dan proses perhitungan akan berhenti. Apabila hasil belum sesuai dengan target yang sudah ditentukan, maka bobot akan diperbarui dan akan dihitung lagi hingga hasil sama dengan target yang sudah ditentukan.
      - Tahap ketiga: mencocokkan hasil *output* dengan pola *output* yang telah dibuat.

c. Hasil
   Hasil dari proses pengenalan pola jenis kendaraan menggunakan metode *single layer perceptron* yaitu sistem bisa mengenali jenis kendaraan roda empat atau roda dua.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

3.4 Pengujian

Pada tahap *testing* dilakukan pengujian untuk mengetahui kinerja sistem. Pengujian dilakukan terhadap 2 citra kendaraan, yaitu kendaraan roda empat dan kendaraan roda dua. Dari hasil pengujian dapat dinilai kemampuan sistem untuk mengenali citra. Dalam mengidentifikasi citra masukan, terlebih dulu citra kendaraan harus melewati beberapa proses agar dapat teridentifikasi dengan baik. Selanjutnya untuk tahap *testing* dibutuhkan nilai *input* dari citra kendaraan yang akan diuji, lalu dilakukan penjumlahan antara bobot akhir kendaraan roda empat dan roda dua. Kemudian dilakukan proses perkalian antara *input* data uji dengan bobot data latih yang sudah disimpan, sehingga dapat dikenali dan ditentukan citra tersebut termasuk kendaraan roda empat atau kendaraan roda dua.

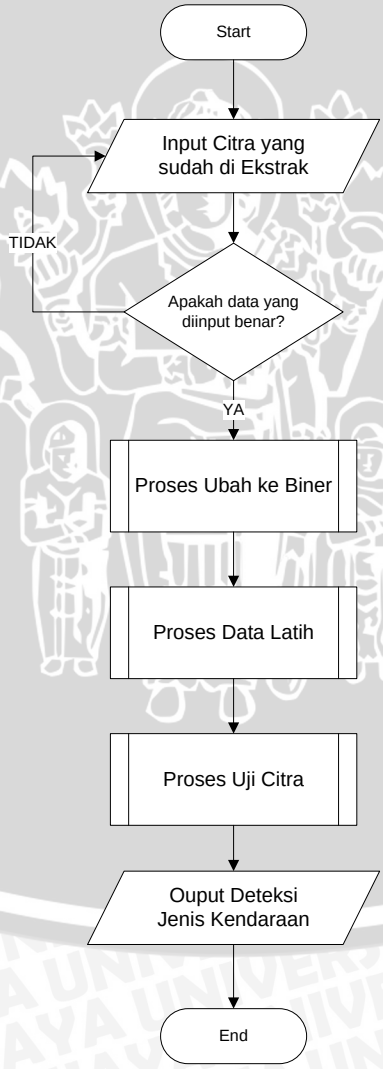
repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

BAB IV PERANCANGAN SISTEM

Pada bab perancangan sistem akan dibahas mengenai perancangan yang akan dibuat, meliputi langkah-langkah penggunaan metode pada implementasi metode *single layer perceptron* untuk deteksi jenis kendaraan roda empat dan roda dua berdasarkan citra *web cam*.

4.1 Perancangan Sistem

Perancangan sistem pada aplikasi yang akan dibuat untuk deteksi kendaraan roda empat dan roda dua, terdiri dari beberapa tahapan antara lain perancangan program aplikasi, perhitungan manual, desain *interface*, dan perancangan pengujian. Pada Gambar 4.1 merupakan *flowchart* sistem secara keseluruhan yang dilakukan pada proses penelitian.

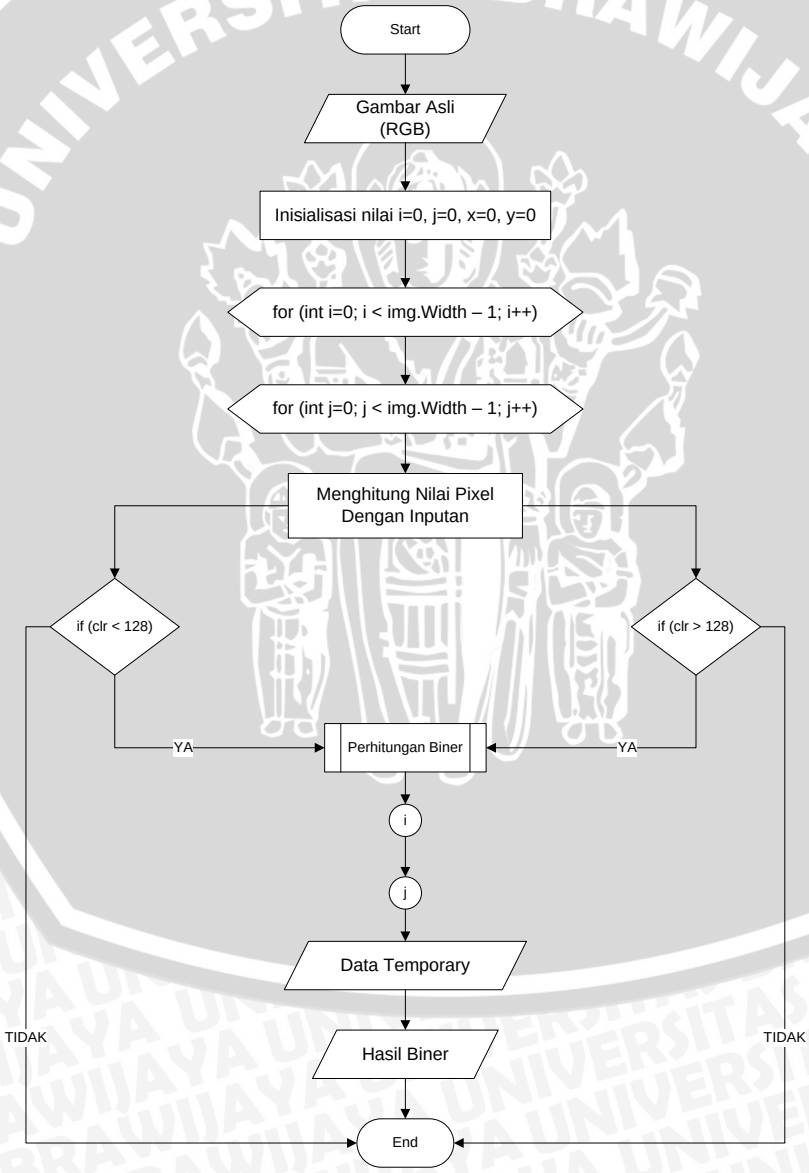


Gambar 4.1 *Flowchart* Sistem Secara Umum

Proses yang dilakukan seperti terlihat pada Gambar 4.1 dengan langkah awal *user* memasukkan data berupa citra dengan format *png*. Setelah itu, gambar asli di ubah menjadi nilai *biner* untuk menghitung proses data latih menggunakan paramater nilai *biner* tersebut. Dalam aplikasi yang akan dibangun proses data latih digunakan untuk menghitung nilai *biner* dari gambar asli kemudian dijadikan sebagai data pembelajaran pada sistem. Setelah dilakukan proses pembelajaran, selanjutnya pengguna melakukan pengujian citra untuk melakukan proses deteksi jenis kendaraan.

4.1.1 Ubah *RGB* Ke *Biner*

Secara keseluruhan proses ubah gambar *RGB* ke *biner* ditunjukkan pada Gambar 4.2.



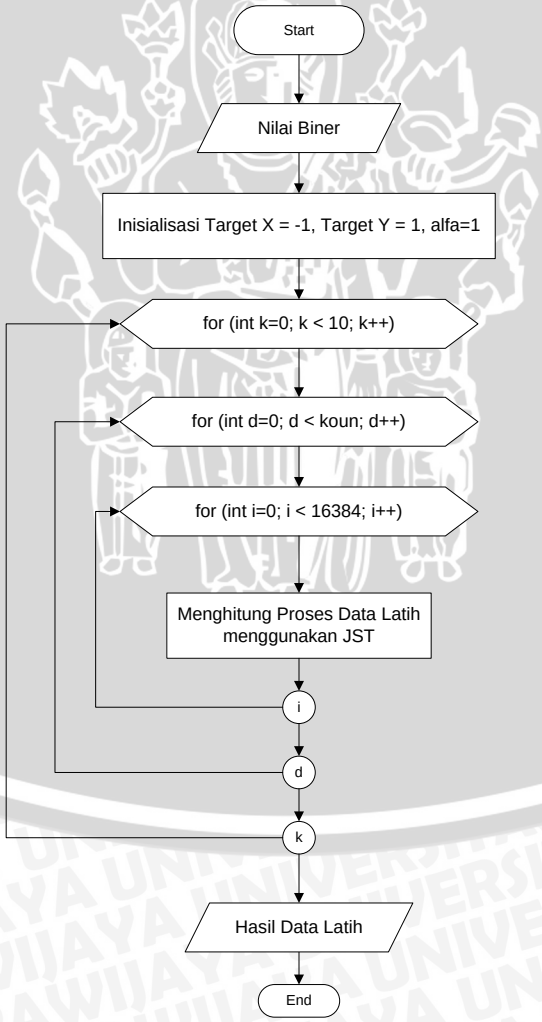
Gambar 4.2 *Flowchart* Ubah *RGB* Ke *Biner*

Adapun proses ubah gambar *RGB* ke *biner* yang ditunjukkan pada Gambar 4.2 dapat dijabarkan sebagai berikut:

1. Input gambar dengan format *.png*.
2. Sistem melakukan proses pengubahan gambar dengan menghitung nilai R, G, dan B dengan rumus (r + g + b) / 3.
3. Setelah nilai biner didapatkan akan ada tes kondisi, dimana jika nilai kurang dari 128 maka sistem memberikan nilai pada gambar 1 sedangkan jika nilai lebih dari 128 sistem memberikan nilai pada gambar -1. Nilai 128 diperoleh dari nilai tengah jumlah maksimum bit, yaitu 256.
4. Data nilai *biner* -1 dan 1 akan disimpan sementara untuk dijadikan nilai X dan Y, dimana nilai X merupakan data biner dari kendaraan roda empat dan nilai Y merupakan data biner dari kendaraan roda dua.

4.1.2 Data Latih

Secara keseluruhan proses data latih dapat dilihat pada Gambar 4.4 berikut.



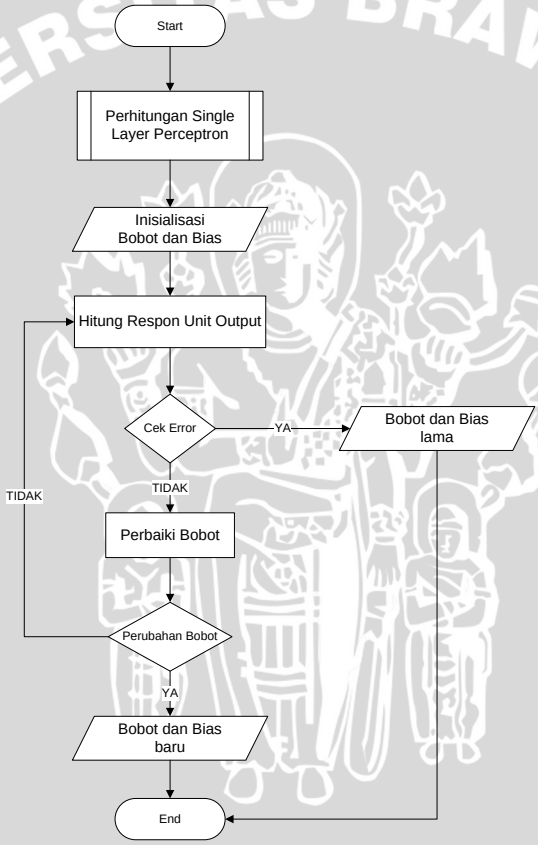
Gambar 4.3 *Flowchart* Data Latih

Proses pada Gambar 4.3 dapat dijabarkan sebagai berikut:

1. Setelah mendapatkan nilai *biner* dari masing-masing *citra* jenis kendaraan roda empat dan roda.
2. Kemudian sistem melakukan proses perhitungan mencari nilai *net, fnet*, bobot, bias, perubahan bobot, dan perubahan bias menggunakan proses perulangan *for*.
3. Setelah itu, menyimpan perubahan nilai bobot akhir.

4.1.3 *Single Layer Perceptron*

Proses metode *single layer perceptron* secara garis besar ditunjukkan pada Gambar 4.4 di bawah ini.



Gambar 4.4 *Flowchart Single Layer Perceptron*

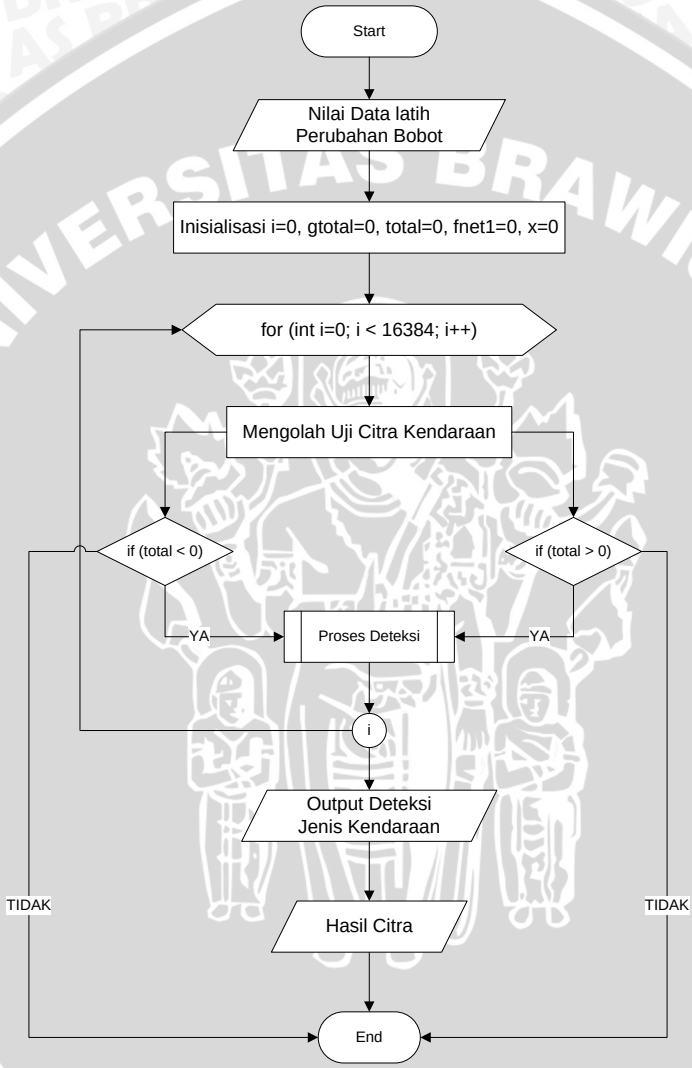
Adapun proses yang dilakukan seperti terlihat pada Gambar 4.4 di atas, yaitu:

1. Langkah awal pada proses tersebut adalah inisialisasi nilai awal bobot dan *bias*, lalu *set* nilai *input* sama dengan vektor *input*.
2. Selanjutnya cek error dengan menghitung respon *unit output*. Jika terjadi error maka bobot dan *bias* harus diperbaiki. Kondisi berhenti *FALSE* jika masih terjadi perubahan bobot dan sebaliknya jika sudah tidak terjadi perubahan bobot maka tes kondisi berhenti adalah *TRUE*. Iterasi akan

dilakukan secara terus menerus hingga *output* sama dengan target yang ditentukan. Modifikasi bobot menggunakan laju pemahaman α (*learning rate*,) yang nilainya dapat diatur.

4.1.4 Uji Citra

Secara keseluruhan proses uji citra dapat dilihat pada Gambar 4.5 seperti di bawah ini.



Gambar 4.5 *Flowchart* Uji Citra

Pada Gambar 4.5 merupakan *flowchart* uji *citra* yang dapat dijabarkan sebagai berikut:

1. Mengolah data perubahan bobot yang didapatkan dari proses data latih.
2. Sistem melakukan proses perhitungan untuk mencari nilai bobot akhir dan menjumlah keseluruhan data.
3. Setelah itu melakukan deteksi jenis kendaraan dengan melakukan kondisi yang didapatkan dari proses perhitungan *single layer perceptron*. Apabila

nilai total -1, maka didapatkan hasil jenis kendaraan roda empat sedangkan jika nilai total 1 maka didapatkan hasil jenis kendaraan roda dua.

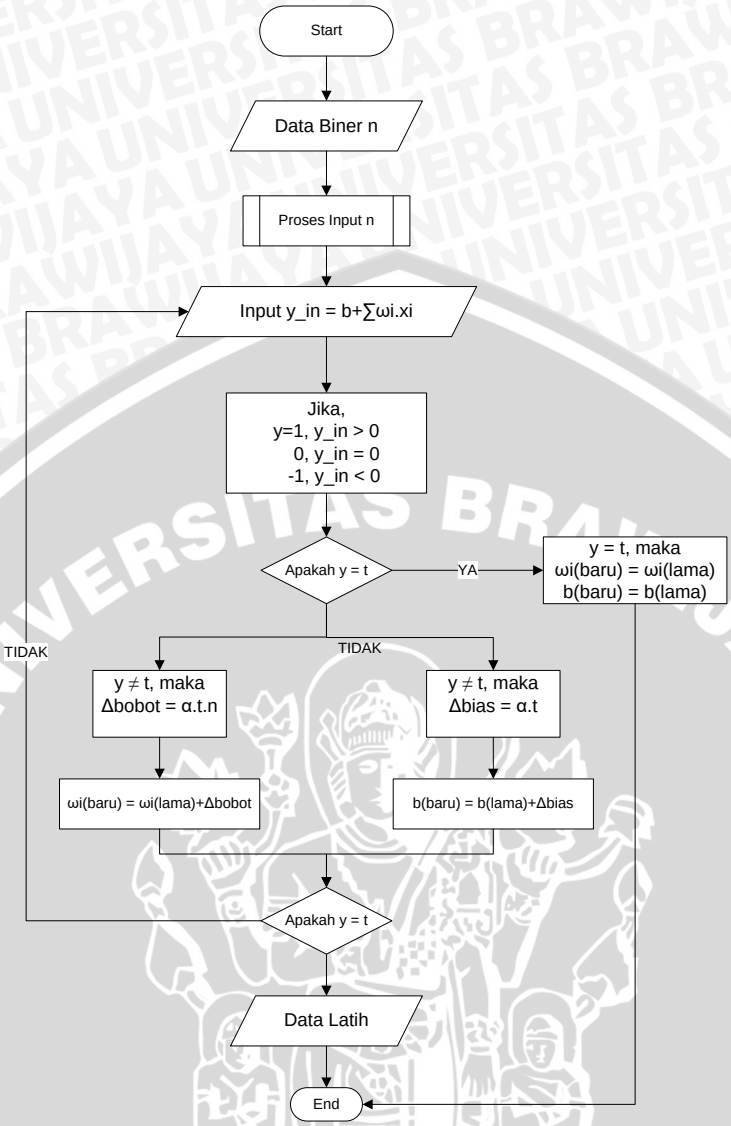
4.2 Perhitungan Manual

Dataset yang digunakan dalam penelitian ini berupa data citra jenis kendaraan berdimensi 25x25 piksel yang terdiri dari kendaraan roda empat dan roda dua seperti yang terdapat pada Lampiran A. Bobot dan bias pada perhitungan manual di *inisialisasi* dengan nilai sebagai berikut, seperti pada Tabel 4.1.

Tabel 4.1 Nilai Bobot, *Bias*, Target, *Learning Rate*

Bobot awal $x\ (\omega_x)$	0
Bobot awal $y\ (\omega_y)$	0
Bias awal $x\ (b_x)$	0
Bias awal $y\ (b_x)$	0
Target $x\ (t_x)$	-1
Target $y\ (t_y)$	1
Learning rate (α)	1

Dalam proses data latih terdapat 3 tahap perhitungan yang diulang selama $y \neq t$, dimana y adalah *fnet* dan t *output*. Proses pertama pada iterasi ini yaitu melakukan perhitungan untuk mencari nilai $y_{in}(net)$ untuk tiap data training. Proses kedua yaitu menghitung respon unit *output* untuk mencari nilai $y(fnet)$ dan proses terakhir melakukan cek kondisi apakah $y = t$. Diagram alir iterasi dapat ditunjukkan pada Gambar 4.6.



Gambar 4.6 Diagram Alir Proses Iterasi

Pada Tabel 4.2 merupakan *input* dari kendaraan roda empat dan roda dua yang sudah di *konversi* ke biner. Data X merupakan nilai biner dari data latih 1 yang berupa kendaraan roda empat dan Y merupakan nilai biner dari data latih 2 yang berupa kendaraan roda dua.

Tabel 4.2 Nilai *Biner*

NO	X	Y
1	-1	-1
2	-1	-1
3	-1	-1
4	-1	-1
5	-1	-1

6	-1	-1
7	-1	-1
8	-1	-1
9	-1	-1
10	-1	-1
11	-1	-1
12	-1	-1
13	-1	-1
14	-1	-1
15	-1	-1
16	-1	-1
17	-1	-1
18	-1	-1
19	-1	-1
20	-1	-1
21	-1	-1
22	-1	-1
23	-1	-1
24	-1	-1
25	-1	-1
26	-1	-1
27	-1	-1
28	-1	-1
29	-1	-1
30	-1	-1
31	-1	-1
32	-1	-1
33	-1	-1
34	-1	-1
35	-1	-1
36	-1	1
37	-1	-1
38	-1	-1
39	-1	1
40	-1	-1

Untuk data dari ke-41 hingga ke-625 dapat dilihat pada Lampiran A. Pada Tabel 4.2 setelah didapatkan nilai *biner*, maka akan di proses untuk perhitungan data latih. Tabel 4.3 hingga Tabel 4.5 merupakan proses data latih kendaraan roda empat epoch ke-1.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Tabel 4.3 Nilai *Net* dan *FNet* Data Latih 1

NO	Y_in(*net*)	Y(*fnet*)
1	0	0
2	0	0
3	0	0
4	0	0
5	0	0
6	0	0
7	0	0
8	0	0
9	0	0
10	0	0
11	0	0
12	0	0
13	0	0
14	0	0
15	0	0
16	0	0
17	0	0
18	0	0
19	0	0
20	0	0
21	0	0
22	0	0
23	0	0
24	0	0
25	0	0
26	0	0
27	0	0
28	0	0
29	0	0
30	0	0
31	0	0
32	0	0
33	0	0
34	0	0
35	0	0
36	0	0
37	0	0
38	0	0
39	0	0
40	0	0

Pada langkah ini dilakukan proses dimana menghitung nilai *net* dan *fnet*. Perhitungan tersebut dilakukan dengan rumus persamaan 2.6. Contoh perhitungan nilai *net* dari tabel 4.3 :

$$y_in = b + \sum_i x_i \omega_i$$

$$y_in = 0 + 0 * (-1)$$

$$y_in = 0$$

Selanjutnya setelah didapatkan nilai *net* sampai data ke 625 seperti di Lampiran A, akan dilakukan hitung respon *unit output* dengan menggunakan persamaan 2.7. Hasil respon *unit output* dari data ke-1 yaitu, 0 $(y_in = 0)$. Karena nilai $fnet \neq t$ maka dilakukan perubahan bobot dengan rumus persamaan 2.8. Contoh perhitungan bobot baru dari Tabel 4.4:

$$\omega_i(baru) = \omega_i(lama) + \alpha * t * x$$

$$\omega_i(baru) = 0 + 1 * (-1) * (-1)$$

$$\omega_i(baru) = 1$$

Tabel 4.4 Perhitungan Bobot dan *Bias* Data Latih 1

NO	Δ BOBOT	Δ BIAS	ω_{baru}	b_{baru}
1	1	-1	1	-1
2	1	-1	1	-1
3	1	-1	1	-1
4	1	-1	1	-1
5	1	-1	1	-1
6	1	-1	1	-1
7	1	-1	1	-1
8	1	-1	1	-1
9	1	-1	1	-1
10	1	-1	1	-1
11	1	-1	1	-1
12	1	-1	1	-1
13	1	-1	1	-1
14	1	-1	1	-1
15	1	-1	1	-1
16	1	-1	1	-1
17	1	-1	1	-1
18	1	-1	1	-1
19	1	-1	1	-1
20	1	-1	1	-1
21	1	-1	1	-1
22	1	-1	1	-1
23	1	-1	1	-1

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

24	1	-1	1	-1
25	1	-1	1	-1
26	1	-1	1	-1
27	1	-1	1	-1
28	1	-1	1	-1
29	1	-1	1	-1
30	1	-1	1	-1
31	1	-1	1	-1
32	1	-1	1	-1
33	1	-1	1	-1
34	1	-1	1	-1
35	1	-1	1	-1
36	1	-1	1	-1
37	1	-1	1	-1
38	1	-1	1	-1
39	1	-1	1	-1
40	1	-1	1	-1

Perhitungan ubah bobot dilakukan sampai data ke-625 seperti pada Lampiran A. Setelah itu akan dilakukan ubah *bias* dengan rumus persamaan 2.9. Contoh perhitungan *bias* baru dari Tabel 4.4:

$$b(baru) = b(lama) + \alpha * t$$

$$b(baru) = 0 + 1*(-1)$$

$$b(baru) = -1$$

Perhitungan *bias* baru dilakukan sampai data ke-625 dan selanjutnya dilakukan proses *epoch* ke-2. Contoh perhitungan *net* dan *fnet* epoch ke-2 Tabel 4.5:

$$y_in = b + \sum_i x_i \omega_i$$

$$y_in = (-1) + 1*(-1)$$

$$y_in = -2$$

Tabel 4.5 Nilai *Net* dan *FNet Epoch* ke-2

NO	Y_in(*net*)	Y(*fnet*)
1	-2	-1
2	-2	-1
3	-2	-1
4	-2	-1
5	-2	-1
6	-2	-1

7	-2	-1
8	-2	-1
9	-2	-1
10	-2	-1
11	-2	-1
12	-2	-1
13	-2	-1
14	-2	-1
15	-2	-1
16	-2	-1
17	-2	-1
18	-2	-1
19	-2	-1
20	-2	-1
21	-2	-1
22	-2	-1
23	-2	-1
24	-2	-1
25	-2	-1
26	-2	-1
27	-2	-1
28	-2	-1
29	-2	-1
30	-2	-1
31	-2	-1
32	-2	-1
33	-2	-1
34	-2	-1
35	-2	-1
36	-2	-1
37	-2	-1
38	-2	-1
39	-2	-1
40	-2	-1

Perhitungan nilai *net* dan *fnet* sampai data ke-625 dapat dilihat pada Lampiran A. Selanjutnya dihitung nilai bobot baru dan *bias* baru, seperti pada Tabel 4.6:

Tabel 4.6 Bobot dan *Bias* Baru *Epoch* ke-2

NO	Δ BOBOT	Δ BIAS	ω_{baru}	b_{baru}
1	0	-1	1	-2
2	0	-1	1	-2
3	0	-1	1	-2

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

4	0	-1	1	-2
5	0	-1	1	-2
6	0	-1	1	-2
7	0	-1	1	-2
8	0	-1	1	-2
9	0	-1	1	-2
10	0	-1	1	-2
11	0	-1	1	-2
12	0	-1	1	-2
13	0	-1	1	-2
14	0	-1	1	-2
15	0	-1	1	-2
16	0	-1	1	-2
17	0	-1	1	-2
18	0	-1	1	-2
19	0	-1	1	-2
20	0	-1	1	-2
21	0	-1	1	-2
22	0	-1	1	-2
23	0	-1	1	-2
24	0	-1	1	-2
25	0	-1	1	-2
26	0	-1	1	-2
27	0	-1	1	-2
28	0	-1	1	-2
29	0	-1	1	-2
30	0	-1	1	-2
31	0	-1	1	-2
32	0	-1	1	-2
33	0	-1	1	-2
34	0	-1	1	-2
35	0	-1	1	-2
36	0	-1	1	-2
37	0	-1	1	-2
38	0	-1	1	-2
39	0	-1	1	-2
40	0	-1	1	-2

Karena nilai *fnet* = *t* maka proses perhitungan selesai, dan selanjutnya dilakukan proses data latih kendaraan roda dua. Tabel 4.7 perhitungan nilai *net* dan *fnet* *epoch* ke-1 kendaraan roda dua.

Tabel 4.7 Nilai *Net* dan *FNet* Data Latih 2

NO	Y_in(*net*)	Y(*fnet*)
1	0	0
2	0	0
3	0	0
4	0	0
5	0	0
6	0	0
7	0	0
8	0	0
9	0	0
10	0	0
11	0	0
12	0	0
13	0	0
14	0	0
15	0	0
16	0	0
17	0	0
18	0	0
19	0	0
20	0	0
21	0	0
22	0	0
23	0	0
24	0	0
25	0	0
26	0	0
27	0	0
28	0	0
29	0	0
30	0	0
31	0	0
32	0	0
33	0	0
34	0	0
35	0	0
36	0	0
37	0	0
38	0	0
39	0	0
40	0	0

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Contoh perhitungan *net* dan *fnet* pada Tabel 4.7:

$$y_in = b + \sum_i x_i \omega_i$$

$$y_in = 0 + 1 * 0$$

$$y_in = 0$$

Hasil respon *unit output* dari data ke-1 yaitu, 0 $(y_in = 0)$. Pada Tabel 4.8 di bawah ini menjelaskan perhitungan bobot dan *bias* baru.

Tabel 4.8 Bobot dan *Bias* Data Latih 2

NO	Δ BOBOT	Δ BIAS	ω_{baru}	b_{baru}
1	-1	1	-1	1
2	-1	1	-1	1
3	-1	1	-1	1
4	-1	1	-1	1
5	-1	1	-1	1
6	-1	1	-1	1
7	-1	1	-1	1
8	-1	1	-1	1
9	-1	1	-1	1
10	-1	1	-1	1
11	-1	1	-1	1
12	-1	1	-1	1
13	-1	1	-1	1
14	-1	1	-1	1
15	-1	1	-1	1
16	-1	1	-1	1
17	-1	1	-1	1
18	-1	1	-1	1
19	-1	1	-1	1
20	-1	1	-1	1
21	-1	1	-1	1
22	-1	1	-1	1
23	-1	1	-1	1
24	-1	1	-1	1
25	-1	1	-1	1
26	-1	1	-1	1
27	-1	1	-1	1
28	-1	1	-1	1
29	-1	1	-1	1
30	-1	1	-1	1

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

31	-1	1	-1	1
32	-1	1	-1	1
33	-1	1	-1	1
34	-1	1	-1	1
35	-1	1	-1	1
36	1	1	1	1
37	-1	1	-1	1
38	-1	1	-1	1
39	1	1	1	1
40	-1	1	-1	1

Ubah bobot dan bias sampai data ke-625 dapat dilihat pada Lampiran A. Berikut contoh perhitungan bobot dan *bias* Tabel 4.8:

$$\omega_i(baru) = \omega_i(lama) + \alpha * t * x$$

$$\omega_i(baru) = 0 + 1*1*(-1)$$

$$\omega_i(baru) = -1$$

$$b_i(baru) = b_i(lama) + \alpha * t$$

$$b_i(baru) = 0 + 1*1$$

$$b_i(baru) = 1$$

Selanjutnya dilakukan proses perhitungan *Epoch* ke-2. Contoh perhitungan nilai *net* dan *fnet epoch* ke-2 Tabel 4.9:

$$y_in = b + \sum_i x_i \omega_i$$

$$y_in = 1 + (-1)*(-1)$$

$$y_in = 2$$

Nilai *FNet* 1, jika $(y_in > 0)$

Tabel 4.9 *Net* dan *FNet Epoch* ke-2

NO	Y_in(*net*)	Y(*fnet*)
1	2	1
2	2	1
3	2	1
4	2	1
5	2	1
6	2	1

7	2	1
8	2	1
9	2	1
10	2	1
11	2	1
12	2	1
13	2	1
14	2	1
15	2	1
16	2	1
17	2	1
18	2	1
19	2	1
20	2	1
21	2	1
22	2	1
23	2	1
24	2	1
25	2	1
26	2	1
27	2	1
28	2	1
29	2	1
30	2	1
31	2	1
32	2	1
33	2	1
34	2	1
35	2	1
36	2	1
37	2	1
38	2	1
39	2	1
40	2	1

Perhitungan *net* dan *fnet* sampai data ke-625 dapat dilihat pada Lampiran A. Selanjutnya dilakukan perhitungan bobot baru dan *bias* baru seperti pada Tabel 4.10.

Tabel 4.10 Bobot dan *Bias Epoch* ke-2

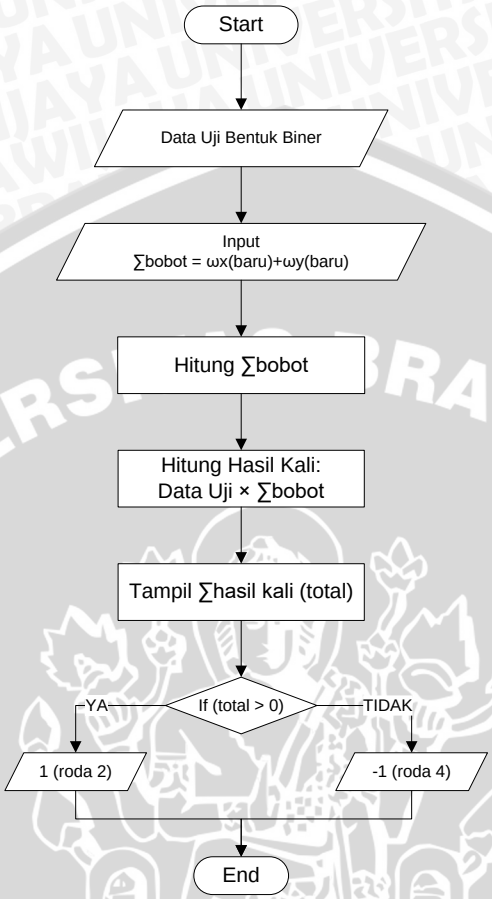
NO	Δ BOBOT	Δ BIAS	ω_{baru}	b_{baru}
1	0	1	-1	2
2	0	1	-1	2
3	0	1	-1	2
4	0	1	-1	2

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

5	0	1	-1	2
6	0	1	-1	2
7	0	1	-1	2
8	0	1	-1	2
9	0	1	-1	2
10	0	1	-1	2
11	0	1	-1	2
12	0	1	-1	2
13	0	1	-1	2
14	0	1	-1	2
15	0	1	-1	2
16	0	1	-1	2
17	0	1	-1	2
18	0	1	-1	2
19	0	1	-1	2
20	0	1	-1	2
21	0	1	-1	2
22	0	1	-1	2
23	0	1	-1	2
24	0	1	-1	2
25	0	1	-1	2
26	0	1	-1	2
27	0	1	-1	2
28	0	1	-1	2
29	0	1	-1	2
30	0	1	-1	2
31	0	1	-1	2
32	0	1	-1	2
33	0	1	-1	2
34	0	1	-1	2
35	0	1	-1	2
36	0	1	1	2
37	0	1	-1	2
38	0	1	-1	2
39	0	1	1	2
40	0	1	-1	2

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Alur perhitungan untuk menentukan kondisi akhir dari suatu Uji Citra diperlukan beberapa langkah yang dapat ditunjukkan pada diagram alir Gambar 4.7 berikut ini.



Gambar 4.7 Diagram Alir Proses Uji Citra

Karena nilai *fnet* = *t* maka proses perhitungan selesai, dan selanjutnya dilakukan uji citra. Pada perhitungan manual ini dilakukan uji citra untuk deteksi kendaraan roda dua. Uji citra dapat dilihat pada Tabel 4.11 berikut ini:

Tabel 4.11 Uji *Citra* Kendaraan Roda Dua

NO	DATA UJI	BOBOT AKHIR	HASIL KALI
1	-1	0	0
2	-1	0	0
3	-1	0	0
4	-1	0	0
5	-1	0	0
6	-1	0	0
7	-1	0	0
8	-1	0	0
9	-1	0	0

10	-1	0	0
11	-1	0	0
12	-1	0	0
13	-1	0	0
14	-1	0	0
15	-1	0	0
16	-1	0	0
17	-1	0	0
18	-1	0	0
19	-1	0	0
20	-1	0	0
21	-1	0	0
22	-1	0	0
23	-1	0	0
24	-1	0	0
25	-1	0	0
26	-1	0	0
27	-1	0	0
28	-1	0	0
29	-1	0	0
30	-1	0	0
31	-1	0	0
32	-1	0	0
33	-1	0	0
34	-1	0	0
35	1	0	0
36	-1	2	-2
37	-1	0	0
38	-1	0	0
39	1	2	2
40	-1	0	0
Total Hasil Akhir			**122**
Kondisi Akhir			**1**

Dimana setelah dilakukan total perhitungan data uji, maka dapat disimpulkan jika kondisi akhir sebagai berikut:

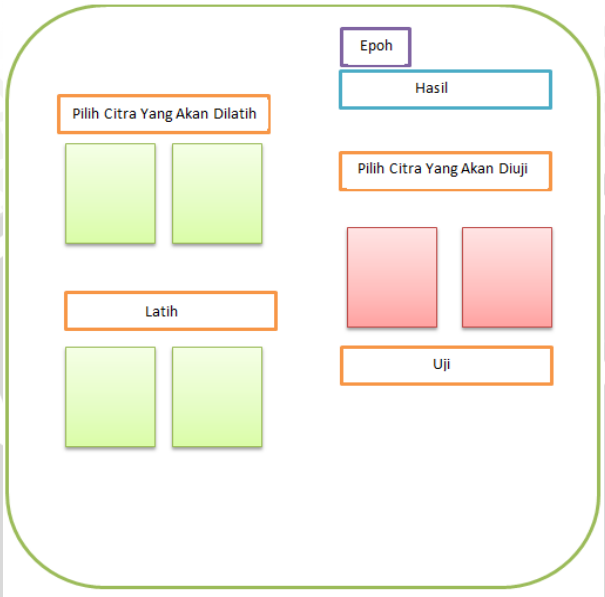
$if \sum total > 0,$ maka nilai 1 yang menentukan jenis kendaraan roda dua

$if \sum total < 0,$ maka nilai -1 yang menentukan jenis kendaraan roda empat

Dari hasil perhitungan manual uji citra, diperoleh hasil akhir 1 yaitu kendaraan roda dua. Untuk perhitungan data uji keseluruhan dapat dilihat pada Lampiran A.

4.3 Desain Interface

Aplikasi deteksi untuk kendaraan roda empat dan roda dua dibuat sederhana untuk memudahkan pengguna. Desain interface untuk aplikasi yang akan dibuat dapat dilihat pada Gambar 4.8 seperti berikut ini.



Gambar 4.8 Desain *Interface* Sistem Deteksi Kendaraan

Keterangan Gambar 4.8 adalah:

1. Tombol pilih citra digunakan untuk memilih kendaraan roda empat dan roda dua yang akan dilatih pada folder yang sudah ditentukan.
2. Tombol latih digunakan untuk melatih citra kendaraan
3. Tombol pilih data uji digunakan untuk memilih kendaraan yang akan di uji pada folder yang sudah ditentukan.
4. Tombol uji digunakan untuk melakukan pengujian.

BAB V IMPLEMENTASI DAN PENGUJIAN

Pada bab ini akan dibahas mengenai hasil implementasi dan pengujian terhadap implementasi dari aplikasi yang dibuat. Implementasi dan uji coba ini bertujuan untuk mengetahui apakah sistem aplikasi telah dapat berjalan sebagaimana mestinya sesuai dengan pengujian yang dilakukan. Pada bab ini juga akan dibahas mengenai *code* program, *fitur* dan tampilan yang terdapat didalam sistem tersebut.

5.1 Implementasi Sistem

Pada tahap ini akan dilakukan pembuatan aplikasi. Dimulai dari pembuatan kode (program) dari *MicrosoftVisual C#* untuk mengimplementasi aplikasi yang telah dibuat. Setelah fungsi-fungsi tersebut sudah bisa dipanggil dan dijalankan, selanjutnya adalah pembuatan *interface* dan penambahan *fitur – fitur*aplikasi. Dan terakhir adalah melakukan pengujian dari sistem aplikasi yang telah dibuat.

5.1.1 Spesifikasi Perangkat Keras

Ruang lingkup perangkat keras adalah perangkat atau *tools* yang diperlukan untuk proses mengimplementasikan sistem. Adapun ruang lingkup perangkat keras yang diperlukan oleh sistem aplikasi ini.

- Processor Intel Core 2 Duo
- Hardisk 160 GB
- Monitor 14
- Keyboard

5.1.2 Spesifikasi Perangkat Lunak

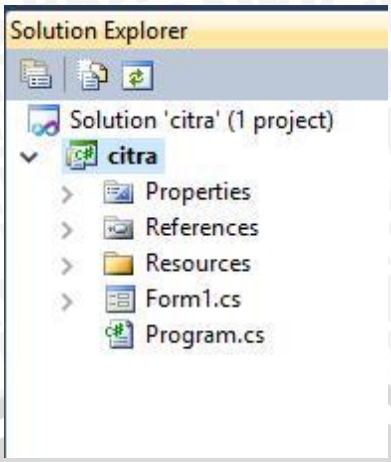
Ruang lingkup perangkat lunak adalah aplikasi atau *software* lain yang diperlukan dalam mengimplementasikan sistem aplikasi. Adapun ruang lingkup perangkat lunak yang diperlukan oleh sistem aplikasi ini antara lain :

- Windows 10
- Microsoft Visual C# 2010

5.2 Source Code Aplikasi

Impementasi aplikasi membahas komponen – komponen pemrograman *(source code)* yang terkait dengan sistem aplikasi yang telah dibuat.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA



Gambar 5.1 Struktur *Source Code Aplikasi*

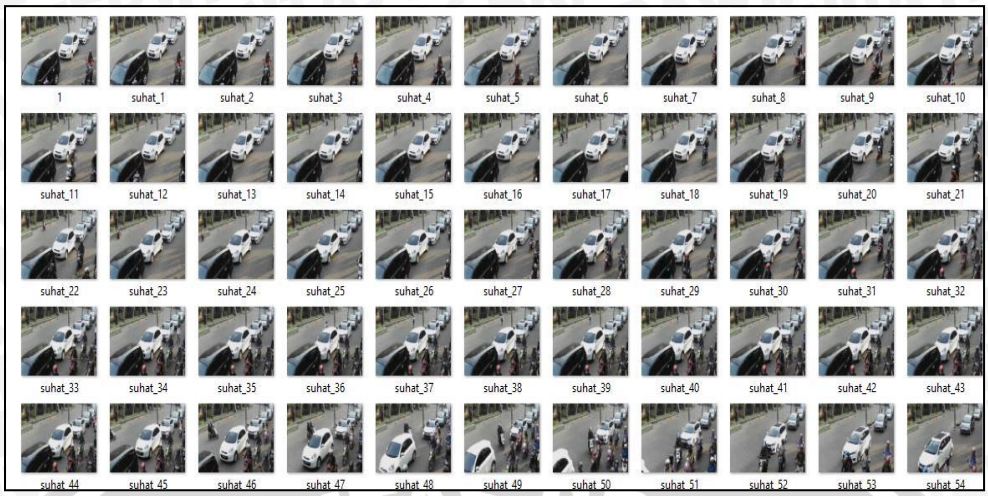
Dapat dilihat pada Gambar 5.1 pada *fileform1.cs* digunakan untuk desain aplikasi. Di dalam file form1.cs terdapat beberapa desain yang meliputi pilih citra, data latih, dan uji citra. Selain itu, penulis menampilkan hasil perhitungan pada tabel. Tabel tersebut meliputi tabel nilai biner x dan y, nilai perhitungan citra kendaraan mobil, dan kendaraan motor. Sedangkan hasil perhitungan pengujian ditampilkan pada tabel nilai perhitungan data uji citra.

5.2.1 Source Code Dan Screenshot Extract Video

```
privatevoid ScanButton_Click(object sender, System.EventArgs e)
{
int lo = 0;
if (md == null) return;
ScanThread st = newScanThread(storagePath, fileName, interval);
do
{
lo = lo + 1;
Thread.Sleep(1000);
label1.Text = "Proses Ekstrak Video : " + counter.ToString();
label1.Invalidate();
label1.Update();
progressBar1.Value = lo;
} while (st.t.IsAlive);
progressBar1.Value = 100;
label1.Text = "Selesai";
}
```

Gambar 5.2 *Source Code* Pilih Citra

Dapat dilihat pada Gambar 5.2 perintah *scanbutton_click* merupakan fungsi untuk melakukan proses ekstrak *video* ke citra. Fungsi perintah kondisi if (md == null) untuk melakukan validasi jika *video* kosong atau belum ada maka sistem otomatis menampilkan *directory*, tetapi jika *video* tidak kosong maka sistem melanjutkan proses berikutnya. Proses berikutnya yaitu, proses ekstrak video menggunakan *library aforge.net* memanggil fungsi yaitu *ScanThread st = new ScanThread(storagePath, fileName, interval)*. Yang dimaksud *storagePath* tempat penyimpanan hasil video yang telah diekstrak, perintah *fileName* untuk membaca nama video yang akan di ekstrak, sedangkan interval merupakan waktu dalam proses ekstrak*video* menjadi *citra*. Hasil ekstrak diubah menjadi citra berekstensi *.jpg* menggunakan *library aforget.net WriteBitmapBitsJpg*.



Gambar 5.3 *Screenshot* Extract *Video* Menjadi *Citra*

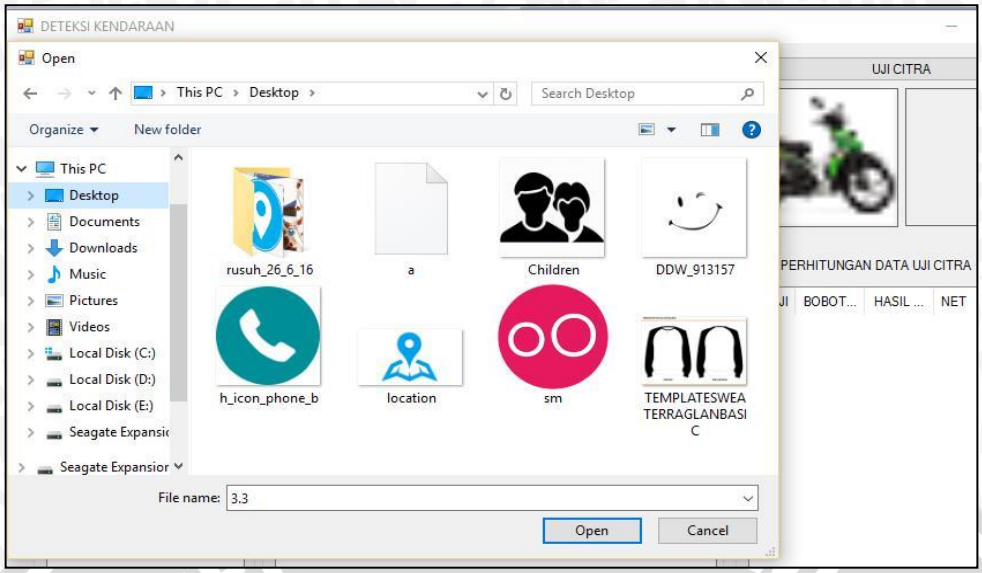
Dapat dilihat *screenshot* pada gambar 5.3, yaitu hasil ekstrak *video* menjadi *citra*dari perintah kode pada gambar 5.2. *Formatcitra* yaitu menggunakan ekstensi *.jpg*. setelah citra diekstrak semua citra akan di convert menjadi ekstensi *.png*. perubahan ini agar dapat dijadikan data pelatihan (*training*) guna untuk melakukan uji coba deteksi kendaraan roda empat maupun roda dua.

5.2.2 Source Code Dan Screenshot Pilih Citra

```
privatevoid button1_Click(object sender, EventArgs e)
{
DialogResult d = openFileDialog1.ShowDialog();
DialogResult a = openFileDialog2.ShowDialog();
if (d == DialogResult.OK || a == DialogResult.OK)
{
pictureBox1.Image = Image.FromFile(openFileDialog1.FileName);
pictureBox2.Image = Image.FromFile(openFileDialog2.FileName);
}}
```

Gambar 5.4 *Source Code* Pilih Citra

Dapat dilihat pada Gambar 5.4 perintah *dialogresult dan openfiledialog* berfungsi untuk menampilkan *directoryfolder* gambar yang akan dicari dan dimasukkan kedalam *formimage*. Sedangkan kondisi *if dialogresult.ok* untuk menangani ketika *user* membatalkan memilih gambar pada *directoryfolder*. Jika gambar dipilih, sistem akan memasukkan gambar tersebut ke dalam *picturebox.image* berdasarkan gambar yang telah dipilih. Dapat dilihat *screenshot* pada Gambar 5.5 dari *source code* pada Gambar 5.4 di bawah ini.



Gambar 5.5 *Screenshot* Pilih Citra

Dapat dilihat *screenshot* pada Gambar 5.5, yaitu *user* memilih gambar untuk dilakukan pengujian data. *Format* gambar yang harus digunakan yaitu *format.png*. Hal ini dikarenakan apabila terdapat warna latar belakang maka sistem kesulitan dalam mengambil objek gambar yang sesungguhnya.

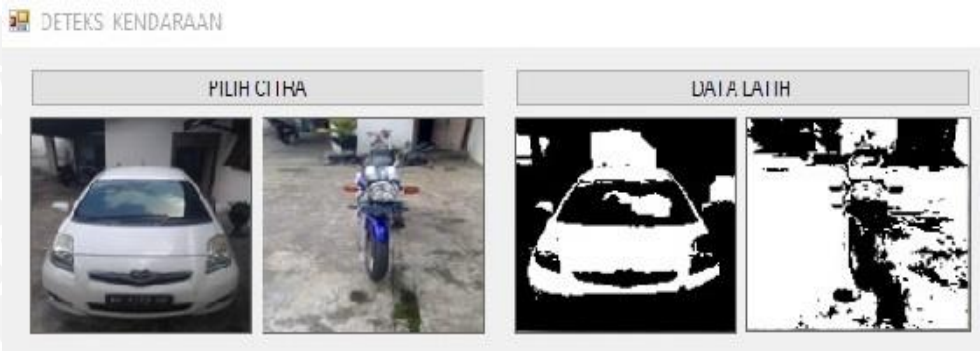
5.2.3 Source Code Dan Screenshot Data Latih

```
Bitmap img1 = newBitmap(pictureBox2.Image);
for (int i = 0; i < img1.Width - 1; i++)
{
for (int j = 0; j < img1.Height - 1; j++)
{
int wh1 = rata_rata(img1.GetPixel(i, j).R, img1.GetPixel(i, j).G,
img1.GetPixel(i, j).B);
if (wh1 < 128)
{
wh1 = 0;
img1.SetPixel(i, j, Color.FromArgb(wh1, wh1, wh1));
}
else
{
wh1 = 255;
img1.SetPixel(i, j, Color.FromArgb(255, wh1, wh1, wh1));
}
}
}
pictureBox3.Image = img;
pictureBox4.Image = img1;
```

Gambar 5.6 *Source Code* Pengubahan Gambar *RGB* menjadi Gambar *Biner*

Dapat dilihat pada gambar 5.6 perintah perulangan *for i* dan *for j* untuk membaca panjang gambar dan tinggi gambar. Setelah itu, mencari nilai rata-rata dengan melakukan penjumlahan *red*, *green*, *blue* dibagi 3. Sedangkan kondisi *if wh1 < 128* untuk memberikan nilai 0 dan 1 pada *pixel* gambar yang akan dilatih. Setelah semua proses tersebut dilakukan, langkah terakhir yaitu melakukan

pengisian gambar dengan menggunakan fungsi *picturebox.image=img*. Dapat dilihat *screenshot* pada Gambar 5.7 dari hasil *source code* pada Gambar 5.6 di



bawah ini.

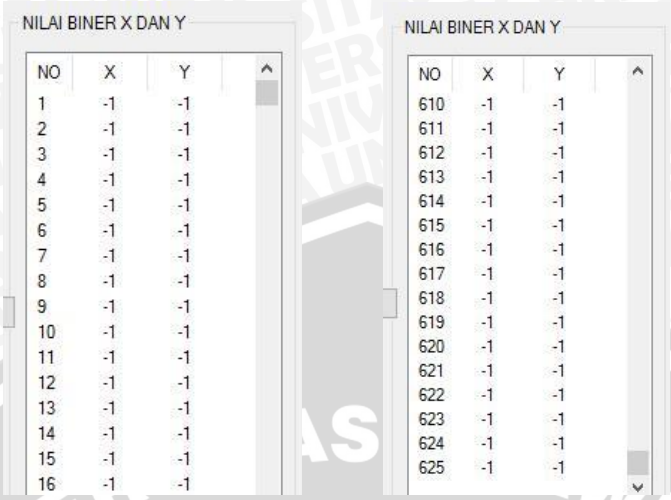
Gambar 5.7 *Screenshot* Pengubahan Gambar *RGB* menjadi Gambar *Biner*

Dapat dilihat *screenshot* pada Gambar 5.7, yaitu perubahan gambar *RGB* menjadi *biner*. Langkah awal yaitu menekan tombol pilih *citra*, kemudian memilih gambar *format.png*. Setelah itu, langkah selanjutnya menekan tombol data latih kemudian sistem melakukan perubahan gambar *RGB* menjadi *biner*. *Source code* selanjutnya dapat dilihat pada Gambar 5.8 berikut.

```
listView1.Clear();
listView1.Columns.Add("NO",30);
listView1.Columns.Add("X",50,HorizontalAlignment.Center);
listView1.Columns.Add("Y",50,HorizontalAlignment.Center);
listView1.View=View.Details;
listView1.FullRowSelect = true;
for (int i = 0; i < temp_x.Count; i++){
ListViewItem lvi = newListViewItem((i+1).ToString());
lvi.SubItems.Add(temp_x[i].ToString());
lvi.SubItems.Add(temp_y[i].ToString());
listView1.Items.Add(lvi);
}
```

Gambar 5.8 *Source Code* Menampilkan Nilai *Biner* Dari Gambar

Dapat dilihat pada Gambar 5.8 perintah *listview* untuk membuat kolom no, *x*, dan *y*. Perulangan *for i* digunakan untuk membaca data nilai *biner* x dan y. Perintah *lvi.subItems.add* untuk menampilkan data ke kolom. Setelah proses itu dilakukan, langkah selanjutnya semua data dimasukan ke dalam tabel nilai *biner* yang sudah disediakan oleh sistem. Dapat dilihat *screenshot* pada Gambar 5.9 dari hasil *source code* pada Gambar 5.8 di bawah ini.



Gambar 5.9 *Screenshot* Tabel Data *Biner* X dan Y

Penjelasan pada Gambar 5.9, yaitu nilai *biner* x dan y. Data *biner* yang digunakan 16384 *pixel* dengan matrik 128 x 128. Nilai *biner* -1 merupakan objek gambar sedangkan nilai *biner* 1 merupakan objek *background.* Variabel x yaitu kendaraan roda empat dan variabel y yaitu kendaraan roda dua. *Source code* selanjutnya dapat dilihat pada Gambar 5.10 di bawah ini.

```
for (int k = 0; k < 10; k++){
for (int d = 0; d < koun ; d++){
if (k != 0){
bobot_awalx = update_bobotx[k];
bias_awalx = update_biasx[k];
bobot_awaly = update_boboty[k];
bias_awaly = update_boboty[k];
}
netx = netx + (Convert.ToInt16(listView1.Items[d].SubItems[1].Text) *
bobot_awalx);
nety = nety + (Convert.ToInt16(listView1.Items[d].SubItems[2].Text) *
bobot_awaly);
}
netx = netx + bias_awalx;
nety = nety + bias_awaly;
if (netx > 0){
fnetx = 1;
}elseif (netx == 0){
fnetx = 0;
}elseif (netx < 0){
fnetx = -1;
}
if (nety > 0){
fnety = 1;
}elseif (nety == 0){
fnety = 0;
}elseif (nety < 0){
fnety = -1;
}
```

Gambar 5.10 *Source Code* Perhitungan *Netx, Nety, Fnetx,* dan *Fnety*

Dapat dilihat pada gambar 5.10 perintah *for k dan for d* untuk melakukan proses perulangan untuk iterasi. Perintah *Convert.ToInt16 (listView1.Items[d].SubItems[1].Text)* untuk melakukan konversi *string* ke *integer*. Setelah itu, dilakukan proses perhitungan *netx* dan *nety*. *Netx* diperoleh dengan melakukan perkalian nilai *biner* x dengan bobot awal x kemudian menjumlahkan hasilnya dengan bias awal. Kondisi *if netx* dan *nety* untuk melakukan perubahan nilai *fnet x* dan *fnet y*, jika *netx* >0 maka sistem memberikan nilai *fnetx* maupun *fnety* yaitu 1. Jika *netx* = 0 maka sistem memberikan nilai *fnetx* maupun *fnety* yaitu 0, dan jika *netx* < 0 maka sistem memberikan nilai *fnetx* maupun *fnety* yaitu -1. Untuk proses pembaharuan bobot dengan perintah *update_bobotx[k]*, *update_boboty[k]*, *update_biasx[k]*, dan *update_biasy[k]* dijelaskan pada *source code* berikutnya. *Source code* selanjutnya dapat dilihat pada Gambar 5.11 di bawah ini.

```
if (Target_X != fnetx || Target_Y != fnety){
delta_bobotx = Target_X * alfa *
Convert.ToInt16(listView1.Items[i].SubItems[1].Text);
delta_biasx = Target_X * alfa * bias1;
delta_boboty = Target_Y * alfa *
Convert.ToInt16(listView1.Items[i].SubItems[2].Text);
delta_biasy = Target_Y * alfa * bias1;
}
else
{
delta_bobotx = 0;
delta_biasx = 0;
delta_boboty = 0;
delta_biasy = 0;
}
rubah_bobotx = bobot_awalx + delta_bobotx;
rubah_biasx = bias_awalx + delta_biasx;
rubah_boboty = bobot_awaly + delta_boboty;
rubah_biasy = bias_awaly + delta_biasy;
```

Gambar 5.11 *Source Code Perhitungan Delta Bobot,Delta Bias, dan Perubahan Bobot*

Dapat dilihat pada gambar 5.11 perintah kondisi *iftargetx* tidak sama dengan *fnetx* maupun *fnety* maka menghitung nilai *delta_bobotx*, *delta_boboty*, *delta_biasx*, dan *delta_biasy*. Perhitungan tersebut menggunakan perintah *target* dikali *alfa* kemudian dikalikan dengan nilai *biner* x dan y. Tetapi jika *targetx* dan *targety*sama dengan *fnetx* dan *fnety* maka sistem memberikan nilai *delta_bobot* dan *delta_bias* yaitu 0. Setelah proses itu dilakukan, selanjutnya melakukan perubahan *bobotx*, *boboty*, *biasx*, dan *biasy* dengan menjumlahkan *(bobot_awalx + delta_bobotx)* dan *(bias_awalx + delta_biasx)*. *Source code* selanjutnya dapat dilihat pada Gambar 5.12 di bawah ini.

```
ListViewItem lv2 = newListViewItem((i + 1).ToString());
lv2.SubItems.Add(netx.ToString());
lv2.SubItems.Add(fnetx.ToString());
lv2.SubItems.Add(delta_bobotx.ToString());
lv2.SubItems.Add(delta_biasx.ToString());
lv2.SubItems.Add(rubah_bobotx.ToString());
```

```
lv2.SubItems.Add(rubah_biasx.ToString());
listView2.Items.Add(lv2);
ListViewItem lv3 = newListViewItem((i + 1).ToString());
lv3.SubItems.Add(nety.ToString());
lv3.SubItems.Add(fnety.ToString());
lv3.SubItems.Add(delta_boboty.ToString());
lv3.SubItems.Add(delta_biasy.ToString());
lv3.SubItems.Add(rubah_boboty.ToString());
lv3.SubItems.Add(rubah_biasy.ToString());
listView3.Items.Add(lv3);
```

Gambar 5.12 *Source Code* Menampilkan Hasil Perhitungan

Dapat dilihat pada Gambar 5.12 perintah *lv2.SubItems.Add* untuk menampilkan data dari hasil perhitungan. Data tersebut meliputi *netx dan nety*, *fnetx* dan *fnety*, *delta_bobotx* dan *delta_boboty*, *delta_biasx* dan *delta_biasy*, *rubah_bobotx* dan *rubah_boboty*, *rubah_biasx* dan *rubah_biasy*. *Source code* selanjutnya dapat dilihat pada Gambar 5.13 di bawah ini.

```
update_bobotx.Add(rubah_bobotx);
update_bobotx.Add(rubah_bobotx);
update_biasx.Add(rubah_biasx);
update_boboty.Add(rubah_boboty);
update_biasy.Add(rubah_biasy);
if ((delta_bobotx == 0 && delta_biasx == 0) || (delta_boboty == 0 &&
delta_biasy == 0)){
a = 1;
}else{
a = 0;
}
}
if (a == 1){
bobot_akhirx.Clear();
bobot_akhiry.Clear();
for (int m = 0; m < koun; m++)
{
bobot_akhirx.Add(update_bobotx[m]);
bobot_akhiry.Add(update_boboty[m]);
}
for (int w = 0;w < koun; w++){
total = bobot_akhirx[w] + bobot_akhiry[w];
bobot_akhirxy.Add(total);
}
return;
}
```

Gambar 5.13 *Source Code* Menyimpan Data Bobot Ke Dalam *List Array*

Dapat dilihat pada Gambar 5.13 perintah *update_bobotx.add*, *update_boboty.add*,*update_biasx.add*,*update_biasy.add* untuk menyimpan nilai perubahan bobot x dan y secara sementara karena bobot ini akan berubah setiap iterasi jika *targetx* maupun *targety* belum sama. Sedangkan untuk menyimpan bobot akhir menggunakan perintah *bobot_akhirx.add* dan *bobot_akhiry.add*. Perintah total menjumlahkan *bobot_akhirx* dan *bobot_akhiry*. Setelah proses tersebut

dilakukan, total disimpan ke dalam data *array* nilai *bobot_akhirxy.add*. Hasil dari perhitungan di atas ditunjukkan dengan Gambar 5.14 dibawah ini.

NILAI PERHITUNGAN CITRA KENDARAAN MOBIL DATA LATIH 1

NO	NET	FNET	D.BOBOT	D.BIAS	R.BOBOT	R.B
72	0	0	1	-1	1	-1
73	0	0	1	-1	1	-1
74	0	0	1	-1	1	-1
75	0	0	1	-1	1	-1
76	0	0	1	-1	1	-1

NILAI PERHITUNGAN CITRA KENDARAAN MOTOR DATA LATIH 2

NO	NET	FNET	D.BOBOT	D.BIAS	R.BOBOT	R.B
1	0	0	-1	1	-1	1
2	0	0	-1	1	-1	1
3	0	0	-1	1	-1	1
4	0	0	-1	1	-1	1
5	0	0	-1	1	-1	1

Gambar 5.14 *Screenshot* Perhitungan Data Latih Citra Kendaraan Mobil Dan Motor

Dapat dilihat pada Gambar 5.14 hasil perhitungan untuk mencari nilai *net*, *fnet*, *delta bobot*, *delta bias*, perubahan bobot, dan perubahan *bias*. Data yang dihitung sebagai data latih yaitu 16384 piksel dengan 2 iterasi. Proses selanjutnya dapat dilihat pada Gambar 5.15 di bawah ini.

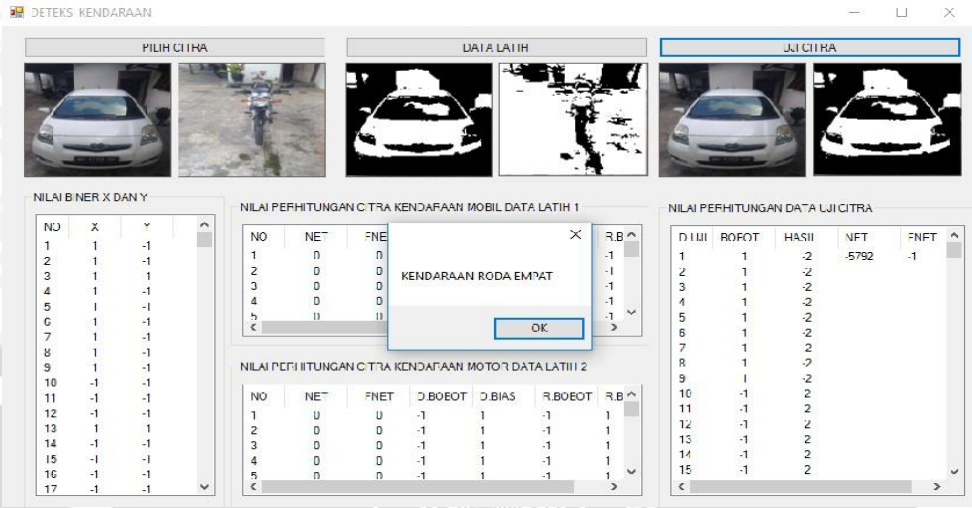
```
for (int i = 0; i < 16384; i++)
{
ListViewItem lv4 = new ListViewItem((i + 1).ToString());
lv4.SubItems.Add(temp_x[i].ToString());

gtotal = temp_x[i] * bobot_akhirxy[i];
lv4.SubItems.Add(gtotal.ToString());
lv4.SubItems.Add("");
lv4.SubItems.Add("");
listView4.Items.Add(lv4);
total = total + gtotal;
}
listView4.Items[0].SubItems[3].Text = total.ToString();
string pesan="";
if (total > 0)

{
fnet1 = 1;
pesan = "KENDARAAN RODA DUA";
}
else if(total < 0)
{
fnet1 = -1;
```

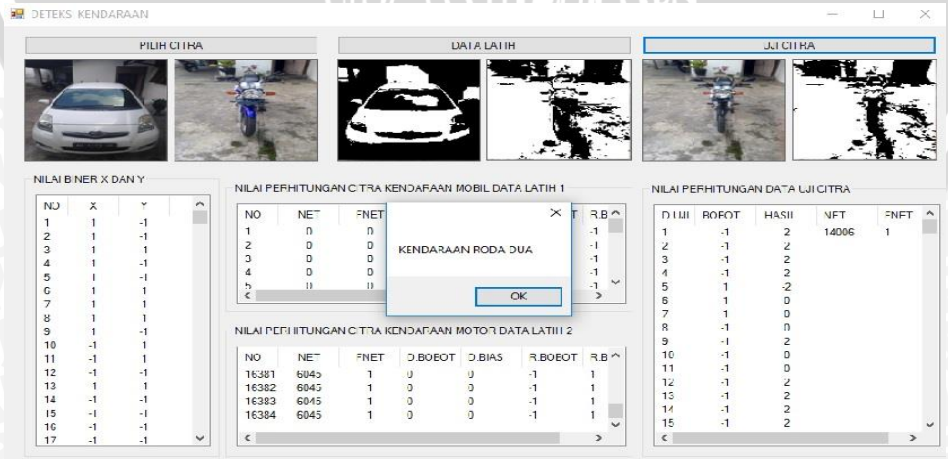
```
pesan = "KENDARAAN RODA EMPAT";
}
            listView4.Items[0].SubItems[4].Text = fnet1.ToString();
            MessageBox.Show(pesan);
                }
```

Gambar 5.15 *Source Code* Uji Citra Deteksi Kendaraan



Gambar 5.16 *Screenshot* Uji Citra Deteksi Kendaraan Roda Empat

Dapat dilihat *screenshot* pada Gambar 5.16, yaitu *user* melakukan uji *citra* deteksi kendaraan roda empat dengan cara memilih data latih berupa *citra* kendaraan roda empat dan roda dua terlebih dahulu. Setelah itu data latih akan diproses dan digunakan sebagai proses pembelajaran agar sistem dapat membedakan kendaraan roda empat dan roda dua. Langkah selanjutnya, *user* melakukan pengujian citra yang telah dipilih yaitu citra kendaraan roda empat kemudian sistem melakukan perhitungan uji citra kendaraan roda empat. Setelah proses tersebut dilakukan dan diperoleh kondisi *fnet* < 1, maka sistem otomatis menampilkan pesan bahwa *citra* tersebut adalah kendaraan roda empat.



Gambar 5.17 *Screenshot* Deteksi Kendaraan Roda Dua

Dapat dilihat *screenshot* pada gambar 5.17, yaitu *user* melakukan uji *citra* deteksi bukan kendaraan roda empat dengan cara memilih data latih berupa *citra* kendaraan roda empat dan roda dua terlebih dahulu. Setelah itu data latih akan diproses dan digunakan untuk proses pembelajaran agar dapat membedakan kendaraan roda empat dan roda dua. Langkah selanjutnya, *user* melakukan pengujian citra yang telah dipilih yaitu citra kendaraan roda roda kemudian sistem melakukan perhitungan uji citra kendaraan roda dua. Setelah proses tersebut dilakukan, maka sistem otomatis menampilkan pesan bahwa *citra* tersebut adalah kendaraan roda dua jika diperoleh kondisi *fnet* > 0.

5.3 Pengujian Sistem

Pengujian sistem dilakukan untuk menguji proses kerja dari aplikasi yang telah dibuat, sehingga aplikasi ini dapat digunakan dengan tepat sesuai fungsinya. Pada pengujian ini digunakan *precision*, *recall*, dan *accuracy* dengan 2 model pengujian yaitu:

- Pengujian untuk mendeteksi kendaraan roda empat
- Pengujian untuk mendeteksi kendaraan
- roda dua

5.3.1 Pengujian Kendaraan Roda Empat

Pada pengujian terhadap kendaraan roda 4 digunakan data uji mobil sebanyak 40, sepeda motor 6, dan bukan keduanya 7. Pengujian akurasi yang telah dibuat dapat dilihat pada Tabel 5.1 di bawah ini:

Tabel 5.1 Pengujian Akurasi Kendaraan Roda Empat

	Mobil	Motor	Bukan Keduanya
Mobil	37	3	0
Motor	1	5	0
Bukan Keduanya	7	0	0

Untuk tabel *confusion matrix* dapat dilihat pada Tabel 5.2 berikut ini.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Tabel 5.2 *Confusion Matrix*

	P	**N**
P	TP 37	FN 3
N	FP 8	TN 5

Precisicion $= \frac{TP}{TP + FP}$

$= \frac{37}{37+8}$

$= 82\%$

Recall $= \frac{TP}{TP + FN}$

$= \frac{37}{37+3}$

$= 92\%$

Accuracy $= \frac{TP + TN}{TP + TN + FP + FN}$

$= \frac{37+5}{37+5+8+3}$

$= \frac{42}{53}$

$= 80\%$

5.3.2 Pengujian Kendaraan Roda Dua

Pada pengujian terhadap kendaraan roda dua digunakan data uji sepeda motor sebanyak 30, mobil 6, dan bukan keduanya 7. Pengujian akurasi yang telah dibuat dapat dilihat pada Tabel 5.3 berikut:

Tabel 5.3 Pengujian Akurasi Kendaraan Roda Dua

	Motor	Mobil	Bukan Keduanya
Motor	27	3	0
Mobil	2	4	0
Bukan Keduanya	0	7	0

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Untuk tabel *confusion matrix* kendaraan roda dua dapat dilihat pada Tabel 5.4 di bawah ini:

Tabel 5.4 *Confusion Matrix*

	P	**N**
P	TP 27	FN 3
N	FP 2	TN 11

Precisicion $= \frac{TP}{TP + FP}$

$= \frac{27}{27 + 2}$

$= 93\%$

Reall $= \frac{TP}{TP + FN}$

$= \frac{27}{27 + 3}$

$= 90\%$

Accuracy $= \frac{TP + TN}{TP + TN + FP + FN}$

$= \frac{27 + 11}{27 + 11 + 3 + 2}$

$= \frac{38}{43}$

$= 88\%$

BAB VI PENUTUP

Setelah dilakukannya implementasi pembuatan dan proses pengujian terhadap aplikasi yang telah dibuat. Ada beberapa hal yang dapat disimpulkan dari pembuatan program ini dan terdapat beberapa saran mengenai apa yang diharapkan dapat digunakan dalam implementasi pembuatan aplikasi ini. Adapun kesimpulan dan saran yang dapat diambil adalah sebagai berikut:

6.1 Kesimpulan

Beberapa kesimpulan yang dapat diambil dari hasil pengujian ini adalah:

1. Sistem mampu mendeteksi kendaraan roda empat dan roda dua dengan *input* berupa *video* yang sudah diekstrak menjadi beberapa *citra* dengan format *bitmap* dan *png*. Pada sistem deteksi kendaraan yang telah dibuat membutuhkan *data training* untuk dijadikan sebagai acuan ketika sistem melakukan pengujian deteksi *citra*.
2. Sistem mampu mengenali objek kendaraan roda empat dengan tingkat akurasi 80% dan kendaraan roda dua dengan akurasi 88%.

6.2 Saran

Adapun saran yang dapat digunakan untuk pengembangan dalam implementasi sistem ini untuk lebih baik, yaitu :

1. Perlu adanya pengembangan menggunakan jaringan syaraf tiruan dengan metode *multilayer* untuk menentukan pola dengan tepat.
2. Perlu adanya penambahan fitur pada aplikasi yang meliputi desain dan menu yang diperlukan.
3. Perlu adanya pengembangan deteksi jenis kendaraan yang berbeda-beda.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

DAFTAR PUSTAKA

Anam, Syamsul. 2008. "*Analisa Kinerja Search Engine Dengan Menilai Precision dan Recall Untuk Informasi Marketing dan Advertising*"

Ansori, Riza, 2009, *"Deteksi Kendaraan Bergerak Secara Realtime"*.

Avianto, Donny, 2016, *"Pengenalan Pola Karakter Plat Nomor Kendaraan Menggunakan Algoritma Momentum Backpropogation Neural Network"*.

Betke, Margrit. Haritaoglu, Esin. Davis, Larry S., 2006, *"Multiple Vehicle Detection and Tracking in Hard Real Time"*.

Care, Duman. Lin, Yu-Feng. and Chen, Tsong-Yi. 2013**.** "*Aplikasi Kamera Pendeteksi Mobil Menggunakan Pendekatan Pengolahan Citra*".

Chen, Thou-Ho. dan Yudi, Made. 2006**.** "*Intelligent Vehicle Counting Method Based on Blob Analysis in Traffic Surveillance*".

Gonzalez, R.C. and R.E Woods. 2002. *Digital Image Processing, 2nd ed*, Prentice-Hall Inc., New Jersey.

Hadi, Setiawan, 2012, "*Deteksi Objek Kendaraan Pada Citra Digital Jalan Raya Menggunakan Metode Visi Computer*".

Hameed, Shihab. Khalifa, Othman. *Et all*, 2010. *Car Monitoring, Alerting and Tracking Model*. International Conference on Computer and Communication Engineering (ICCCE).

Herman, 2015, "*Pengenalan Spesies Gulma Berdasarkan Bentuk dan Tekstur Daun Menggunakan Jaringan Syaraf Triruan*".

Indarawati. 2013. *Egmentasi Citra X-Ray Dari Citra CT Menggunakan Active Contour*. Program Studi Teknik Informatika Jurusan Teknik Elektro, Politeknik Negeri Lhokseumawe.

Kusumadewi, Sri. 2003. *Artificial Intelegence*. Yogyakarta: penerbit Graha Ilmu.

Pujiyanta, Ardi, 2009, *"Pengenalan Objek Sederhana Dengan Jaringan Saraf Tiruan Metode Perceptron"*.

Santi, Candra, 2011, *"Mengubah Citra Berwarna Menajadi Grayscale dan Citra Biner"*.

Sun, Zehang. Bebis, George. and Miller, Ronald., 2006. *On-Road Vehicle Detection: A Review*. IEEE Transactions On Pattern Analysis and Machine Intelligence.

Sutoyo. Mulyanto. Suhartono. Nurhayati. Wijanarko. 2009. *Teori Pengolahan Citra Digital*. Yogyakarta: Penerbit Andi.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

LAMPIRAN A PERHITUNGAN MANUAL

ωx (bobot x)	y (bobot y)	bias awal x	bias awal y	target x	target y	α
0	0	0	0	-1	1	1

NO	X	Y
1	-1	-1
2	-1	-1
3	-1	-1
4	-1	-1
5	-1	-1
6	-1	-1
7	-1	-1
8	-1	-1
9	-1	-1
10	-1	-1
11	-1	-1
12	-1	-1
13	-1	-1
14	-1	-1
15	-1	-1
16	-1	-1
17	-1	-1
18	-1	-1
19	-1	-1
20	-1	-1
21	-1	-1
22	-1	-1
23	-1	-1
24	-1	-1
25	-1	-1
26	-1	-1
27	-1	-1
28	-1	-1
29	-1	-1
30	-1	-1
31	-1	-1
32	-1	-1
33	-1	-1
34	-1	-1
35	-1	-1
36	-1	1
37	-1	-1
38	-1	-1
39	-1	1
40	-1	-1
41	-1	-1

42	-1	-1
43	-1	-1
44	-1	-1
45	-1	-1
46	-1	-1
47	-1	-1
48	-1	-1
49	-1	-1
50	-1	-1
51	-1	-1
52	-1	-1
53	-1	-1
54	-1	-1
55	-1	-1
56	-1	-1
57	-1	1
58	-1	-1
59	-1	-1
60	-1	1
61	-1	1
62	-1	-1
63	-1	1
64	-1	-1
65	1	-1
66	1	1

67	1	1
68	1	-1
69	1	-1
70	-1	-1
71	-1	-1
72	-1	-1
73	-1	-1
74	-1	-1
75	-1	-1
76	-1	-1
77	-1	-1
78	-1	-1
79	-1	-1
80	-1	-1
81	-1	-1
82	-1	1
83	-1	1
84	-1	-1
85	-1	1
86	-1	-1
87	-1	1
88	-1	-1
89	-1	-1
90	-1	1
91	1	1

92	1	-1
93	1	-1
94	-1	1
95	1	-1
96	1	-1
97	-1	-1
98	-1	-1
99	-1	-1
100	-1	-1
101	-1	-1
102	-1	-1
103	-1	-1
104	-1	-1
105	-1	-1
106	-1	-1
107	-1	1
108	-1	1
109	-1	1
110	-1	-1
111	-1	1
112	-1	-1
113	1	-1
114	-1	1
115	-1	1
116	-1	1

117	1	1
118	1	-1
119	-1	1
120	1	-1
121	1	-1
122	-1	-1
123	-1	-1
124	-1	-1
125	-1	-1
126	-1	-1
127	-1	-1
128	-1	-1
129	-1	-1
130	-1	-1
131	-1	-1
132	-1	1
133	-1	1
134	-1	1
135	1	1
136	-1	-1
137	-1	1
138	-1	1
139	-1	1
140	-1	-1
141	-1	1

142	1	1
143	1	-1
144	-1	1
145	1	-1
146	1	-1
147	-1	-1
148	-1	-1
149	-1	-1
150	-1	-1
151	-1	-1
152	-1	-1
153	-1	-1
154	-1	-1
155	-1	-1
156	-1	-1
157	1	-1
158	-1	1
159	-1	1
160	1	1
161	-1	-1
162	-1	1
163	-1	1
164	-1	1
165	-1	1
166	-1	1

167	1	1
168	1	1
169	1	1
170	1	-1
171	-1	-1
172	-1	-1
173	-1	-1
174	-1	-1
175	-1	-1
176	-1	-1
177	-1	-1
178	-1	-1
179	-1	-1
180	1	-1
181	-1	-1
182	1	-1
183	-1	1
184	1	1
185	1	1
186	-1	1
187	1	-1
188	-1	1
189	-1	1
190	-1	1
191	1	-1

192	1	1
193	1	1
194	1	-1
195	1	-1
196	-1	-1
197	-1	-1
198	-1	-1
199	-1	-1
200	-1	-1
201	-1	-1
202	-1	1
203	-1	-1
204	1	-1
205	-1	-1
206	-1	-1
207	1	-1
208	-1	1
209	1	1
210	-1	1
211	-1	1
212	1	-1
213	-1	1
214	-1	1
215	1	1
216	1	1

217	1	1
218	1	1
219	1	-1
220	1	-1
221	1	-1
222	-1	-1
223	-1	-1
224	-1	-1
225	-1	-1
226	-1	-1
227	-1	1
228	1	1
229	-1	-1
230	-1	-1
231	-1	-1
232	1	-1
233	-1	-1
234	-1	1
235	-1	1
236	-1	1
237	-1	1
238	-1	-1
239	-1	1
240	1	1
241	1	-1

242	-1	1
243	-1	1
244	-1	-1
245	1	-1
246	1	-1
247	-1	-1
248	-1	-1
249	-1	-1
250	-1	-1
251	-1	-1
252	-1	-1
253	-1	-1
254	-1	-1
255	1	-1
256	1	-1
257	1	-1
258	-1	-1
259	-1	1
260	-1	1
261	-1	1
262	-1	1
263	-1	-1
264	1	1
265	1	1
266	1	-1

267	1	1
268	-1	1
269	1	-1
270	1	-1
271	1	-1
272	-1	-1
273	-1	-1
274	-1	-1
275	-1	-1
276	-1	-1
277	1	-1
278	1	-1
279	1	-1
280	-1	1
281	-1	-1
282	1	-1
283	-1	-1
284	-1	-1
285	-1	1
286	-1	-1
287	-1	-1
288	-1	-1
289	-1	-1
290	-1	-1
291	1	1

292	-1	1
293	-1	1
294	-1	-1
295	-1	-1
296	1	-1
297	1	-1
298	1	-1
299	-1	-1
300	-1	-1
301	-1	-1
302	1	-1
303	1	-1
304	-1	-1
305	1	1
306	1	1
307	1	-1
308	-1	-1
309	-1	-1
310	-1	-1
311	-1	-1
312	1	-1
313	-1	1
314	-1	1
315	-1	1
316	1	-1

317	1	-1
318	1	1
319	1	-1
320	1	-1
321	1	-1
322	1	-1
323	1	-1
324	1	-1
325	-1	-1
326	-1	-1
327	1	-1
328	1	-1
329	1	-1
330	1	1
331	1	1
332	1	-1
333	-1	-1
334	-1	-1
335	-1	-1
336	-1	-1
337	-1	-1
338	-1	1
339	1	1
340	1	1
341	1	1

342	1	1
343	1	-1
344	-1	-1
345	1	-1
346	-1	-1
347	-1	-1
348	1	-1
349	1	-1
350	-1	-1
351	-1	-1
352	1	-1
353	1	-1
354	-1	-1
355	1	-1
356	-1	1
357	-1	1
358	1	-1
359	-1	-1
360	-1	-1
361	1	-1
362	1	1
363	1	1
364	1	1
365	1	1
366	1	1

367	1	1
368	1	1
369	1	-1
370	1	-1
371	1	-1
372	1	-1
373	-1	-1
374	-1	-1
375	-1	-1
376	-1	-1
377	1	-1
378	1	-1
379	1	1
380	1	-1
381	1	1
382	1	1
383	1	-1
384	-1	-1
385	-1	-1
386	1	-1
387	1	1
388	1	1
389	-1	1
390	-1	1
391	1	1

392	1	1
393	1	1
394	1	-1
395	-1	-1
396	-1	-1
397	-1	-1
398	-1	-1
399	-1	-1
400	-1	-1
401	-1	-1
402	-1	-1
403	-1	-1
404	-1	1
405	1	1
406	-1	-1
407	1	1
408	-1	-1
409	-1	-1
410	1	-1
411	1	1
412	1	1
413	1	1
414	-1	1
415	-1	1
416	-1	1

417	-1	1
418	1	1
419	-1	1
420	-1	-1
421	-1	-1
422	-1	-1
423	-1	-1
424	-1	-1
425	-1	-1
426	-1	-1
427	1	-1
428	-1	-1
429	1	-1
430	1	1
431	1	1
432	1	-1
433	-1	-1
434	-1	-1
435	1	-1
436	1	1
437	1	1
438	-1	1
439	-1	1
440	-1	1
441	-1	1

442	1	1
443	-1	-1
444	-1	-1
445	-1	1
446	-1	1
447	-1	-1
448	-1	-1
449	-1	-1
450	-1	-1
451	-1	-1
452	1	1
453	-1	-1
454	1	-1
455	1	-1
456	1	-1
457	-1	-1
458	-1	-1
459	1	-1
460	1	-1
461	1	1
462	1	1
463	-1	1
464	-1	1
465	-1	-1
466	1	-1

467	-1	1
468	-1	1
469	-1	-1
470	-1	-1
471	-1	-1
472	-1	1
473	-1	-1
474	-1	-1
475	-1	-1
476	-1	-1
477	1	-1
478	-1	-1
479	-1	-1
480	-1	-1
481	1	-1
482	-1	-1
483	-1	-1
484	1	-1
485	1	-1
486	1	1
487	-1	1
488	-1	1
489	-1	-1
490	1	1
491	1	-1

492	-1	-1
493	-1	1
494	-1	-1
495	-1	-1
496	-1	-1
497	-1	1
498	-1	-1
499	-1	-1
500	-1	-1
501	-1	-1
502	-1	-1
503	-1	-1
504	1	-1
505	1	-1
506	1	-1
507	-1	-1
508	-1	-1
509	1	-1
510	1	-1
511	-1	-1
512	-1	1
513	-1	1
514	1	1
515	1	-1
516	1	-1

517	1	-1
518	1	-1
519	-1	-1
520	-1	-1
521	-1	-1
522	-1	-1
523	-1	1
524	-1	-1
525	-1	-1
526	-1	-1
527	-1	-1
528	-1	-1
529	-1	-1
530	1	-1
531	-1	-1
532	-1	-1
533	1	-1
534	1	-1
535	-1	-1
536	1	-1
537	1	1
538	1	1
539	1	1
540	1	-1
541	-1	-1

542	-1	-1
543	-1	-1
544	-1	-1
545	-1	-1
546	-1	-1
547	-1	1
548	-1	1
549	-1	-1
550	-1	-1
551	-1	-1
552	-1	-1
553	-1	-1
554	-1	-1
555	-1	-1
556	-1	-1
557	-1	-1
558	-1	-1
559	-1	-1
560	-1	-1
561	-1	-1
562	-1	-1
563	-1	1
564	-1	-1
565	-1	1
566	-1	1

567	-1	-1
568	-1	-1
569	-1	-1
570	-1	-1
571	-1	1
572	-1	1
573	-1	-1
574	-1	-1
575	-1	-1
576	-1	-1
577	-1	-1
578	-1	-1
579	-1	-1
580	-1	-1
581	-1	-1
582	-1	-1
583	-1	-1
584	-1	-1
585	-1	-1
586	-1	-1
587	-1	-1
588	-1	-1
589	-1	-1
590	-1	-1
591	-1	1

592	-1	1
593	-1	1
594	-1	1
595	-1	1
596	-1	1
597	-1	-1
598	-1	-1
599	-1	-1
600	-1	-1
601	-1	-1
602	-1	-1
603	-1	-1
604	-1	-1
605	-1	-1
606	-1	-1
607	-1	-1
608	-1	-1

609	-1	-1
610	-1	-1
611	-1	-1
612	-1	-1
613	-1	-1
614	-1	-1
615	-1	-1
616	-1	-1
617	-1	-1
618	-1	-1
619	-1	-1
620	-1	-1
621	-1	-1
622	-1	-1
623	-1	-1
624	-1	-1
625	-1	-1

KENDARAAN RODA EMPAT

NO	y_in(*net*)	y(*fnet*)	Δ BOBOT	Δ BIAS	ω_{baru}	b_{baru}
1	0	0	1	-1	1	-1
2	0	0	1	-1	1	-1
3	0	0	1	-1	1	-1
4	0	0	1	-1	1	-1
5	0	0	1	-1	1	-1
6	0	0	1	-1	1	-1
7	0	0	1	-1	1	-1
8	0	0	1	-1	1	-1
9	0	0	1	-1	1	-1
10	0	0	1	-1	1	-1
11	0	0	1	-1	1	-1
12	0	0	1	-1	1	-1
13	0	0	1	-1	1	-1
14	0	0	1	-1	1	-1
15	0	0	1	-1	1	-1
16	0	0	1	-1	1	-1
17	0	0	1	-1	1	-1
18	0	0	1	-1	1	-1
19	0	0	1	-1	1	-1
20	0	0	1	-1	1	-1
21	0	0	1	-1	1	-1
22	0	0	1	-1	1	-1

NO	y_in(*net*)	y(*fnet*)	Δ BOBOT	Δ BIAS	ω_{baru}	b_{baru}
1	-2	-1	0	-1	1	-2
2	-2	-1	0	-1	1	-2
3	-2	-1	0	-1	1	-2
4	-2	-1	0	-1	1	-2
5	-2	-1	0	-1	1	-2
6	-2	-1	0	-1	1	-2
7	-2	-1	0	-1	1	-2
8	-2	-1	0	-1	1	-2
9	-2	-1	0	-1	1	-2
10	-2	-1	0	-1	1	-2
11	-2	-1	0	-1	1	-2
12	-2	-1	0	-1	1	-2
13	-2	-1	0	-1	1	-2
14	-2	-1	0	-1	1	-2
15	-2	-1	0	-1	1	-2
16	-2	-1	0	-1	1	-2
17	-2	-1	0	-1	1	-2
18	-2	-1	0	-1	1	-2
19	-2	-1	0	-1	1	-2
20	-2	-1	0	-1	1	-2
21	-2	-1	0	-1	1	-2
22	-2	-1	0	-1	1	-2

23	0	0	1	-1	1	-1
24	0	0	1	-1	1	-1
25	0	0	1	-1	1	-1
26	0	0	1	-1	1	-1
27	0	0	1	-1	1	-1
28	0	0	1	-1	1	-1
29	0	0	1	-1	1	-1
30	0	0	1	-1	1	-1
31	0	0	1	-1	1	-1
32	0	0	1	-1	1	-1
33	0	0	1	-1	1	-1
34	0	0	1	-1	1	-1
35	0	0	1	-1	1	-1
36	0	0	1	-1	1	-1
37	0	0	1	-1	1	-1
38	0	0	1	-1	1	-1
39	0	0	1	-1	1	-1
40	0	0	1	-1	1	-1
41	0	0	1	-1	1	-1
42	0	0	1	-1	1	-1
43	0	0	1	-1	1	-1
44	0	0	1	-1	1	-1
45	0	0	1	-1	1	-1
46	0	0	1	-1	1	-1
47	0	0	1	-1	1	-1

23	-2	-1	0	-1	1	-2
24	-2	-1	0	-1	1	-2
25	-2	-1	0	-1	1	-2
26	-2	-1	0	-1	1	-2
27	-2	-1	0	-1	1	-2
28	-2	-1	0	-1	1	-2
29	-2	-1	0	-1	1	-2
30	-2	-1	0	-1	1	-2
31	-2	-1	0	-1	1	-2
32	-2	-1	0	-1	1	-2
33	-2	-1	0	-1	1	-2
34	-2	-1	0	-1	1	-2
35	-2	-1	0	-1	1	-2
36	-2	-1	0	-1	1	-2
37	-2	-1	0	-1	1	-2
38	-2	-1	0	-1	1	-2
39	-2	-1	0	-1	1	-2
40	-2	-1	0	-1	1	-2
41	-2	-1	0	-1	1	-2
42	-2	-1	0	-1	1	-2
43	-2	-1	0	-1	1	-2
44	-2	-1	0	-1	1	-2
45	-2	-1	0	-1	1	-2
46	-2	-1	0	-1	1	-2
47	-2	-1	0	-1	1	-2

48	0	0	1	-1	1	-1
49	0	0	1	-1	1	-1
50	0	0	1	-1	1	-1
51	0	0	1	-1	1	-1
52	0	0	1	-1	1	-1
53	0	0	1	-1	1	-1
54	0	0	1	-1	1	-1
55	0	0	1	-1	1	-1
56	0	0	1	-1	1	-1
57	0	0	1	-1	1	-1
58	0	0	1	-1	1	-1
59	0	0	1	-1	1	-1
60	0	0	1	-1	1	-1
61	0	0	1	-1	1	-1
62	0	0	1	-1	1	-1
63	0	0	1	-1	1	-1
64	0	0	1	-1	1	-1
65	0	0	-1	-1	-1	-1
66	0	0	-1	-1	-1	-1
67	0	0	-1	-1	-1	-1
68	0	0	-1	-1	-1	-1
69	0	0	-1	-1	-1	-1
70	0	0	1	-1	1	-1
71	0	0	1	-1	1	-1
72	0	0	1	-1	1	-1

48	-2	-1	0	-1	1	-2
49	-2	-1	0	-1	1	-2
50	-2	-1	0	-1	1	-2
51	-2	-1	0	-1	1	-2
52	-2	-1	0	-1	1	-2
53	-2	-1	0	-1	1	-2
54	-2	-1	0	-1	1	-2
55	-2	-1	0	-1	1	-2
56	-2	-1	0	-1	1	-2
57	-2	-1	0	-1	1	-2
58	-2	-1	0	-1	1	-2
59	-2	-1	0	-1	1	-2
60	-2	-1	0	-1	1	-2
61	-2	-1	0	-1	1	-2
62	-2	-1	0	-1	1	-2
63	-2	-1	0	-1	1	-2
64	-2	-1	0	-1	1	-2
65	-2	-1	0	-1	-1	-2
66	-2	-1	0	-1	-1	-2
67	-2	-1	0	-1	-1	-2
68	-2	-1	0	-1	-1	-2
69	-2	-1	0	-1	-1	-2
70	-2	-1	0	-1	1	-2
71	-2	-1	0	-1	1	-2
72	-2	-1	0	-1	1	-2

73	0	0	1	-1	1	-1
74	0	0	1	-1	1	-1
75	0	0	1	-1	1	-1
76	0	0	1	-1	1	-1
77	0	0	1	-1	1	-1
78	0	0	1	-1	1	-1
79	0	0	1	-1	1	-1
80	0	0	1	-1	1	-1
81	0	0	1	-1	1	-1
82	0	0	1	-1	1	-1
83	0	0	1	-1	1	-1
84	0	0	1	-1	1	-1
85	0	0	1	-1	1	-1
86	0	0	1	-1	1	-1
87	0	0	1	-1	1	-1
88	0	0	1	-1	1	-1
89	0	0	1	-1	1	-1
90	0	0	1	-1	1	-1
91	0	0	-1	-1	-1	-1
92	0	0	-1	-1	-1	-1
93	0	0	-1	-1	-1	-1
94	0	0	1	-1	1	-1
95	0	0	-1	-1	-1	-1
96	0	0	-1	-1	-1	-1
97	0	0	1	-1	1	-1

73	-2	-1	0	-1	1	-2
74	-2	-1	0	-1	1	-2
75	-2	-1	0	-1	1	-2
76	-2	-1	0	-1	1	-2
77	-2	-1	0	-1	1	-2
78	-2	-1	0	-1	1	-2
79	-2	-1	0	-1	1	-2
80	-2	-1	0	-1	1	-2
81	-2	-1	0	-1	1	-2
82	-2	-1	0	-1	1	-2
83	-2	-1	0	-1	1	-2
84	-2	-1	0	-1	1	-2
85	-2	-1	0	-1	1	-2
86	-2	-1	0	-1	1	-2
87	-2	-1	0	-1	1	-2
88	-2	-1	0	-1	1	-2
89	-2	-1	0	-1	1	-2
90	-2	-1	0	-1	1	-2
91	-2	-1	0	-1	-1	-2
92	-2	-1	0	-1	-1	-2
93	-2	-1	0	-1	-1	-2
94	-2	-1	0	-1	1	-2
95	-2	-1	0	-1	-1	-2
96	-2	-1	0	-1	-1	-2
97	-2	-1	0	-1	1	-2

98	0	0	1	-1	1	-1
99	0	0	1	-1	1	-1
100	0	0	1	-1	1	-1
101	0	0	1	-1	1	-1
102	0	0	1	-1	1	-1
103	0	0	1	-1	1	-1
104	0	0	1	-1	1	-1
105	0	0	1	-1	1	-1
106	0	0	1	-1	1	-1
107	0	0	1	-1	1	-1
108	0	0	1	-1	1	-1
109	0	0	1	-1	1	-1
110	0	0	1	-1	1	-1
111	0	0	1	-1	1	-1
112	0	0	1	-1	1	-1
113	0	0	-1	-1	-1	-1
114	0	0	1	-1	1	-1
115	0	0	1	-1	1	-1
116	0	0	1	-1	1	-1
117	0	0	-1	-1	-1	-1
118	0	0	-1	-1	-1	-1
119	0	0	1	-1	1	-1
120	0	0	-1	-1	-1	-1
121	0	0	-1	-1	-1	-1
122	0	0	1	-1	1	-1

98	-2	-1	0	-1	1	-2
99	-2	-1	0	-1	1	-2
100	-2	-1	0	-1	1	-2
101	-2	-1	0	-1	1	-2
102	-2	-1	0	-1	1	-2
103	-2	-1	0	-1	1	-2
104	-2	-1	0	-1	1	-2
105	-2	-1	0	-1	1	-2
106	-2	-1	0	-1	1	-2
107	-2	-1	0	-1	1	-2
108	-2	-1	0	-1	1	-2
109	-2	-1	0	-1	1	-2
110	-2	-1	0	-1	1	-2
111	-2	-1	0	-1	1	-2
112	-2	-1	0	-1	1	-2
113	-2	-1	0	-1	-1	-2
114	-2	-1	0	-1	1	-2
115	-2	-1	0	-1	1	-2
116	-2	-1	0	-1	1	-2
117	-2	-1	0	-1	-1	-2
118	-2	-1	0	-1	-1	-2
119	-2	-1	0	-1	1	-2
120	-2	-1	0	-1	-1	-2
121	-2	-1	0	-1	-1	-2
122	-2	-1	0	-1	1	-2

123	0	0	1	-1	1	-1
124	0	0	1	-1	1	-1
125	0	0	1	-1	1	-1
126	0	0	1	-1	1	-1
127	0	0	1	-1	1	-1
128	0	0	1	-1	1	-1
129	0	0	1	-1	1	-1
130	0	0	1	-1	1	-1
131	0	0	1	-1	1	-1
132	0	0	1	-1	1	-1
133	0	0	1	-1	1	-1
134	0	0	1	-1	1	-1
135	0	0	-1	-1	-1	-1
136	0	0	1	-1	1	-1
137	0	0	1	-1	1	-1
138	0	0	1	-1	1	-1
139	0	0	1	-1	1	-1
140	0	0	1	-1	1	-1
141	0	0	1	-1	1	-1
142	0	0	-1	-1	-1	-1
143	0	0	-1	-1	-1	-1
144	0	0	1	-1	1	-1
145	0	0	-1	-1	-1	-1
146	0	0	-1	-1	-1	-1
147	0	0	1	-1	1	-1

123	-2	-1	0	-1	1	-2
124	-2	-1	0	-1	1	-2
125	-2	-1	0	-1	1	-2
126	-2	-1	0	-1	1	-2
127	-2	-1	0	-1	1	-2
128	-2	-1	0	-1	1	-2
129	-2	-1	0	-1	1	-2
130	-2	-1	0	-1	1	-2
131	-2	-1	0	-1	1	-2
132	-2	-1	0	-1	1	-2
133	-2	-1	0	-1	1	-2
134	-2	-1	0	-1	1	-2
135	-2	-1	0	-1	-1	-2
136	-2	-1	0	-1	1	-2
137	-2	-1	0	-1	1	-2
138	-2	-1	0	-1	1	-2
139	-2	-1	0	-1	1	-2
140	-2	-1	0	-1	1	-2
141	-2	-1	0	-1	1	-2
142	-2	-1	0	-1	-1	-2
143	-2	-1	0	-1	-1	-2
144	-2	-1	0	-1	1	-2
145	-2	-1	0	-1	-1	-2
146	-2	-1	0	-1	-1	-2
147	-2	-1	0	-1	1	-2

148	0	0	1	-1	1	-1
149	0	0	1	-1	1	-1
150	0	0	1	-1	1	-1
151	0	0	1	-1	1	-1
152	0	0	1	-1	1	-1
153	0	0	1	-1	1	-1
154	0	0	1	-1	1	-1
155	0	0	1	-1	1	-1
156	0	0	1	-1	1	-1
157	0	0	-1	-1	-1	-1
158	0	0	1	-1	1	-1
159	0	0	1	-1	1	-1
160	0	0	-1	-1	-1	-1
161	0	0	1	-1	1	-1
162	0	0	1	-1	1	-1
163	0	0	1	-1	1	-1
164	0	0	1	-1	1	-1
165	0	0	1	-1	1	-1
166	0	0	1	-1	1	-1
167	0	0	-1	-1	-1	-1
168	0	0	-1	-1	-1	-1
169	0	0	-1	-1	-1	-1
170	0	0	-1	-1	-1	-1
171	0	0	1	-1	1	-1
172	0	0	1	-1	1	-1

148	-2	-1	0	-1	1	-2
149	-2	-1	0	-1	1	-2
150	-2	-1	0	-1	1	-2
151	-2	-1	0	-1	1	-2
152	-2	-1	0	-1	1	-2
153	-2	-1	0	-1	1	-2
154	-2	-1	0	-1	1	-2
155	-2	-1	0	-1	1	-2
156	-2	-1	0	-1	1	-2
157	-2	-1	0	-1	-1	-2
158	-2	-1	0	-1	1	-2
159	-2	-1	0	-1	1	-2
160	-2	-1	0	-1	-1	-2
161	-2	-1	0	-1	1	-2
162	-2	-1	0	-1	1	-2
163	-2	-1	0	-1	1	-2
164	-2	-1	0	-1	1	-2
165	-2	-1	0	-1	1	-2
166	-2	-1	0	-1	1	-2
167	-2	-1	0	-1	-1	-2
168	-2	-1	0	-1	-1	-2
169	-2	-1	0	-1	-1	-2
170	-2	-1	0	-1	-1	-2
171	-2	-1	0	-1	1	-2
172	-2	-1	0	-1	1	-2

173	0	0	1	-1	1	-1
174	0	0	1	-1	1	-1
175	0	0	1	-1	1	-1
176	0	0	1	-1	1	-1
177	0	0	1	-1	1	-1
178	0	0	1	-1	1	-1
179	0	0	1	-1	1	-1
180	0	0	-1	-1	-1	-1
181	0	0	1	-1	1	-1
182	0	0	-1	-1	-1	-1
183	0	0	1	-1	1	-1
184	0	0	-1	-1	-1	-1
185	0	0	-1	-1	-1	-1
186	0	0	1	-1	1	-1
187	0	0	-1	-1	-1	-1
188	0	0	1	-1	1	-1
189	0	0	1	-1	1	-1
190	0	0	1	-1	1	-1
191	0	0	-1	-1	-1	-1
192	0	0	-1	-1	-1	-1
193	0	0	-1	-1	-1	-1
194	0	0	-1	-1	-1	-1
195	0	0	-1	-1	-1	-1
196	0	0	1	-1	1	-1
197	0	0	1	-1	1	-1

173	-2	-1	0	-1	1	-2
174	-2	-1	0	-1	1	-2
175	-2	-1	0	-1	1	-2
176	-2	-1	0	-1	1	-2
177	-2	-1	0	-1	1	-2
178	-2	-1	0	-1	1	-2
179	-2	-1	0	-1	1	-2
180	-2	-1	0	-1	-1	-2
181	-2	-1	0	-1	1	-2
182	-2	-1	0	-1	-1	-2
183	-2	-1	0	-1	1	-2
184	-2	-1	0	-1	-1	-2
185	-2	-1	0	-1	-1	-2
186	-2	-1	0	-1	1	-2
187	-2	-1	0	-1	-1	-2
188	-2	-1	0	-1	1	-2
189	-2	-1	0	-1	1	-2
190	-2	-1	0	-1	1	-2
191	-2	-1	0	-1	-1	-2
192	-2	-1	0	-1	-1	-2
193	-2	-1	0	-1	-1	-2
194	-2	-1	0	-1	-1	-2
195	-2	-1	0	-1	-1	-2
196	-2	-1	0	-1	1	-2
197	-2	-1	0	-1	1	-2

198	0	0	1	-1	1	-1
199	0	0	1	-1	1	-1
200	0	0	1	-1	1	-1
201	0	0	1	-1	1	-1
202	0	0	1	-1	1	-1
203	0	0	1	-1	1	-1
204	0	0	-1	-1	-1	-1
205	0	0	1	-1	1	-1
206	0	0	1	-1	1	-1
207	0	0	-1	-1	-1	-1
208	0	0	1	-1	1	-1
209	0	0	-1	-1	-1	-1
210	0	0	1	-1	1	-1
211	0	0	1	-1	1	-1
212	0	0	-1	-1	-1	-1
213	0	0	1	-1	1	-1
214	0	0	1	-1	1	-1
215	0	0	-1	-1	-1	-1
216	0	0	-1	-1	-1	-1
217	0	0	-1	-1	-1	-1
218	0	0	-1	-1	-1	-1
219	0	0	-1	-1	-1	-1
220	0	0	-1	-1	-1	-1
221	0	0	-1	-1	-1	-1
222	0	0	1	-1	1	-1

198	-2	-1	0	-1	1	-2
199	-2	-1	0	-1	1	-2
200	-2	-1	0	-1	1	-2
201	-2	-1	0	-1	1	-2
202	-2	-1	0	-1	1	-2
203	-2	-1	0	-1	1	-2
204	-2	-1	0	-1	-1	-2
205	-2	-1	0	-1	1	-2
206	-2	-1	0	-1	1	-2
207	-2	-1	0	-1	-1	-2
208	-2	-1	0	-1	1	-2
209	-2	-1	0	-1	-1	-2
210	-2	-1	0	-1	1	-2
211	-2	-1	0	-1	1	-2
212	-2	-1	0	-1	-1	-2
213	-2	-1	0	-1	1	-2
214	-2	-1	0	-1	1	-2
215	-2	-1	0	-1	-1	-2
216	-2	-1	0	-1	-1	-2
217	-2	-1	0	-1	-1	-2
218	-2	-1	0	-1	-1	-2
219	-2	-1	0	-1	-1	-2
220	-2	-1	0	-1	-1	-2
221	-2	-1	0	-1	-1	-2
222	-2	-1	0	-1	1	-2

223	0	0	1	-1	1	-1
224	0	0	1	-1	1	-1
225	0	0	1	-1	1	-1
226	0	0	1	-1	1	-1
227	0	0	1	-1	1	-1
228	0	0	-1	-1	-1	-1
229	0	0	1	-1	1	-1
230	0	0	1	-1	1	-1
231	0	0	1	-1	1	-1
232	0	0	-1	-1	-1	-1
233	0	0	1	-1	1	-1
234	0	0	1	-1	1	-1
235	0	0	1	-1	1	-1
236	0	0	1	-1	1	-1
237	0	0	1	-1	1	-1
238	0	0	1	-1	1	-1
239	0	0	1	-1	1	-1
240	0	0	-1	-1	-1	-1
241	0	0	-1	-1	-1	-1
242	0	0	1	-1	1	-1
243	0	0	1	-1	1	-1
244	0	0	1	-1	1	-1
245	0	0	-1	-1	-1	-1
246	0	0	-1	-1	-1	-1
247	0	0	1	-1	1	-1

223	-2	-1	0	-1	1	-2
224	-2	-1	0	-1	1	-2
225	-2	-1	0	-1	1	-2
226	-2	-1	0	-1	1	-2
227	-2	-1	0	-1	1	-2
228	-2	-1	0	-1	-1	-2
229	-2	-1	0	-1	1	-2
230	-2	-1	0	-1	1	-2
231	-2	-1	0	-1	1	-2
232	-2	-1	0	-1	-1	-2
233	-2	-1	0	-1	1	-2
234	-2	-1	0	-1	1	-2
235	-2	-1	0	-1	1	-2
236	-2	-1	0	-1	1	-2
237	-2	-1	0	-1	1	-2
238	-2	-1	0	-1	1	-2
239	-2	-1	0	-1	1	-2
240	-2	-1	0	-1	-1	-2
241	-2	-1	0	-1	-1	-2
242	-2	-1	0	-1	1	-2
243	-2	-1	0	-1	1	-2
244	-2	-1	0	-1	1	-2
245	-2	-1	0	-1	-1	-2
246	-2	-1	0	-1	-1	-2
247	-2	-1	0	-1	1	-2

248	0	0	1	-1	1	-1
249	0	0	1	-1	1	-1
250	0	0	1	-1	1	-1
251	0	0	1	-1	1	-1
252	0	0	1	-1	1	-1
253	0	0	1	-1	1	-1
254	0	0	1	-1	1	-1
255	0	0	-1	-1	-1	-1
256	0	0	-1	-1	-1	-1
257	0	0	-1	-1	-1	-1
258	0	0	1	-1	1	-1
259	0	0	1	-1	1	-1
260	0	0	1	-1	1	-1
261	0	0	1	-1	1	-1
262	0	0	1	-1	1	-1
263	0	0	1	-1	1	-1
264	0	0	-1	-1	-1	-1
265	0	0	-1	-1	-1	-1
266	0	0	-1	-1	-1	-1
267	0	0	-1	-1	-1	-1
268	0	0	1	-1	1	-1
269	0	0	-1	-1	-1	-1
270	0	0	-1	-1	-1	-1
271	0	0	-1	-1	-1	-1
272	0	0	1	-1	1	-1

248	-2	-1	0	-1	1	-2
249	-2	-1	0	-1	1	-2
250	-2	-1	0	-1	1	-2
251	-2	-1	0	-1	1	-2
252	-2	-1	0	-1	1	-2
253	-2	-1	0	-1	1	-2
254	-2	-1	0	-1	1	-2
255	-2	-1	0	-1	-1	-2
256	-2	-1	0	-1	-1	-2
257	-2	-1	0	-1	-1	-2
258	-2	-1	0	-1	1	-2
259	-2	-1	0	-1	1	-2
260	-2	-1	0	-1	1	-2
261	-2	-1	0	-1	1	-2
262	-2	-1	0	-1	1	-2
263	-2	-1	0	-1	1	-2
264	-2	-1	0	-1	-1	-2
265	-2	-1	0	-1	-1	-2
266	-2	-1	0	-1	-1	-2
267	-2	-1	0	-1	-1	-2
268	-2	-1	0	-1	1	-2
269	-2	-1	0	-1	-1	-2
270	-2	-1	0	-1	-1	-2
271	-2	-1	0	-1	-1	-2
272	-2	-1	0	-1	1	-2

273	0	0	1	-1	1	-1
274	0	0	1	-1	1	-1
275	0	0	1	-1	1	-1
276	0	0	1	-1	1	-1
277	0	0	-1	-1	-1	-1
278	0	0	-1	-1	-1	-1
279	0	0	-1	-1	-1	-1
280	0	0	1	-1	1	-1
281	0	0	1	-1	1	-1
282	0	0	-1	-1	-1	-1
283	0	0	1	-1	1	-1
284	0	0	1	-1	1	-1
285	0	0	1	-1	1	-1
286	0	0	1	-1	1	-1
287	0	0	1	-1	1	-1
288	0	0	1	-1	1	-1
289	0	0	1	-1	1	-1
290	0	0	1	-1	1	-1
291	0	0	-1	-1	-1	-1
292	0	0	1	-1	1	-1
293	0	0	1	-1	1	-1
294	0	0	1	-1	1	-1
295	0	0	1	-1	1	-1
296	0	0	-1	-1	-1	-1
297	0	0	-1	-1	-1	-1

273	-2	-1	0	-1	1	-2
274	-2	-1	0	-1	1	-2
275	-2	-1	0	-1	1	-2
276	-2	-1	0	-1	1	-2
277	-2	-1	0	-1	-1	-2
278	-2	-1	0	-1	-1	-2
279	-2	-1	0	-1	-1	-2
280	-2	-1	0	-1	1	-2
281	-2	-1	0	-1	1	-2
282	-2	-1	0	-1	-1	-2
283	-2	-1	0	-1	1	-2
284	-2	-1	0	-1	1	-2
285	-2	-1	0	-1	1	-2
286	-2	-1	0	-1	1	-2
287	-2	-1	0	-1	1	-2
288	-2	-1	0	-1	1	-2
289	-2	-1	0	-1	1	-2
290	-2	-1	0	-1	1	-2
291	-2	-1	0	-1	-1	-2
292	-2	-1	0	-1	1	-2
293	-2	-1	0	-1	1	-2
294	-2	-1	0	-1	1	-2
295	-2	-1	0	-1	1	-2
296	-2	-1	0	-1	-1	-2
297	-2	-1	0	-1	-1	-2

298	0	0	-1	-1	-1	-1
299	0	0	1	-1	1	-1
300	0	0	1	-1	1	-1
301	0	0	1	-1	1	-1
302	0	0	-1	-1	-1	-1
303	0	0	-1	-1	-1	-1
304	0	0	1	-1	1	-1
305	0	0	-1	-1	-1	-1
306	0	0	-1	-1	-1	-1
307	0	0	-1	-1	-1	-1
308	0	0	1	-1	1	-1
309	0	0	1	-1	1	-1
310	0	0	1	-1	1	-1
311	0	0	1	-1	1	-1
312	0	0	-1	-1	-1	-1
313	0	0	1	-1	1	-1
314	0	0	1	-1	1	-1
315	0	0	1	-1	1	-1
316	0	0	-1	-1	-1	-1
317	0	0	-1	-1	-1	-1
318	0	0	-1	-1	-1	-1
319	0	0	-1	-1	-1	-1
320	0	0	-1	-1	-1	-1
321	0	0	-1	-1	-1	-1
322	0	0	-1	-1	-1	-1

298	-2	-1	0	-1	-1	-2
299	-2	-1	0	-1	1	-2
300	-2	-1	0	-1	1	-2
301	-2	-1	0	-1	1	-2
302	-2	-1	0	-1	-1	-2
303	-2	-1	0	-1	-1	-2
304	-2	-1	0	-1	1	-2
305	-2	-1	0	-1	-1	-2
306	-2	-1	0	-1	-1	-2
307	-2	-1	0	-1	-1	-2
308	-2	-1	0	-1	1	-2
309	-2	-1	0	-1	1	-2
310	-2	-1	0	-1	1	-2
311	-2	-1	0	-1	1	-2
312	-2	-1	0	-1	-1	-2
313	-2	-1	0	-1	1	-2
314	-2	-1	0	-1	1	-2
315	-2	-1	0	-1	1	-2
316	-2	-1	0	-1	-1	-2
317	-2	-1	0	-1	-1	-2
318	-2	-1	0	-1	-1	-2
319	-2	-1	0	-1	-1	-2
320	-2	-1	0	-1	-1	-2
321	-2	-1	0	-1	-1	-2
322	-2	-1	0	-1	-1	-2

323	0	0	-1	-1	-1	-1
324	0	0	-1	-1	-1	-1
325	0	0	1	-1	1	-1
326	0	0	1	-1	1	-1
327	0	0	-1	-1	-1	-1
328	0	0	-1	-1	-1	-1
329	0	0	-1	-1	-1	-1
330	0	0	-1	-1	-1	-1
331	0	0	-1	-1	-1	-1
332	0	0	-1	-1	-1	-1
333	0	0	1	-1	1	-1
334	0	0	1	-1	1	-1
335	0	0	1	-1	1	-1
336	0	0	1	-1	1	-1
337	0	0	1	-1	1	-1
338	0	0	1	-1	1	-1
339	0	0	-1	-1	-1	-1
340	0	0	-1	-1	-1	-1
341	0	0	-1	-1	-1	-1
342	0	0	-1	-1	-1	-1
343	0	0	-1	-1	-1	-1
344	0	0	1	-1	1	-1
345	0	0	-1	-1	-1	-1
346	0	0	1	-1	1	-1
347	0	0	1	-1	1	-1

323	-2	-1	0	-1	-1	-2
324	-2	-1	0	-1	-1	-2
325	-2	-1	0	-1	1	-2
326	-2	-1	0	-1	1	-2
327	-2	-1	0	-1	-1	-2
328	-2	-1	0	-1	-1	-2
329	-2	-1	0	-1	-1	-2
330	-2	-1	0	-1	-1	-2
331	-2	-1	0	-1	-1	-2
332	-2	-1	0	-1	-1	-2
333	-2	-1	0	-1	1	-2
334	-2	-1	0	-1	1	-2
335	-2	-1	0	-1	1	-2
336	-2	-1	0	-1	1	-2
337	-2	-1	0	-1	1	-2
338	-2	-1	0	-1	1	-2
339	-2	-1	0	-1	-1	-2
340	-2	-1	0	-1	-1	-2
341	-2	-1	0	-1	-1	-2
342	-2	-1	0	-1	-1	-2
343	-2	-1	0	-1	-1	-2
344	-2	-1	0	-1	1	-2
345	-2	-1	0	-1	-1	-2
346	-2	-1	0	-1	1	-2
347	-2	-1	0	-1	1	-2

348	0	0	-1	-1	-1	-1
349	0	0	-1	-1	-1	-1
350	0	0	1	-1	1	-1
351	0	0	1	-1	1	-1
352	0	0	-1	-1	-1	-1
353	0	0	-1	-1	-1	-1
354	0	0	1	-1	1	-1
355	0	0	-1	-1	-1	-1
356	0	0	1	-1	1	-1
357	0	0	1	-1	1	-1
358	0	0	-1	-1	-1	-1
359	0	0	1	-1	1	-1
360	0	0	1	-1	1	-1
361	0	0	-1	-1	-1	-1
362	0	0	-1	-1	-1	-1
363	0	0	-1	-1	-1	-1
364	0	0	-1	-1	-1	-1
365	0	0	-1	-1	-1	-1
366	0	0	-1	-1	-1	-1
367	0	0	-1	-1	-1	-1
368	0	0	-1	-1	-1	-1
369	0	0	-1	-1	-1	-1
370	0	0	-1	-1	-1	-1
371	0	0	-1	-1	-1	-1
372	0	0	-1	-1	-1	-1

348	-2	-1	0	-1	-1	-2
349	-2	-1	0	-1	-1	-2
350	-2	-1	0	-1	1	-2
351	-2	-1	0	-1	1	-2
352	-2	-1	0	-1	-1	-2
353	-2	-1	0	-1	-1	-2
354	-2	-1	0	-1	1	-2
355	-2	-1	0	-1	-1	-2
356	-2	-1	0	-1	1	-2
357	-2	-1	0	-1	1	-2
358	-2	-1	0	-1	-1	-2
359	-2	-1	0	-1	1	-2
360	-2	-1	0	-1	1	-2
361	-2	-1	0	-1	-1	-2
362	-2	-1	0	-1	-1	-2
363	-2	-1	0	-1	-1	-2
364	-2	-1	0	-1	-1	-2
365	-2	-1	0	-1	-1	-2
366	-2	-1	0	-1	-1	-2
367	-2	-1	0	-1	-1	-2
368	-2	-1	0	-1	-1	-2
369	-2	-1	0	-1	-1	-2
370	-2	-1	0	-1	-1	-2
371	-2	-1	0	-1	-1	-2
372	-2	-1	0	-1	-1	-2

373	0	0	1	-1	1	-1
374	0	0	1	-1	1	-1
375	0	0	1	-1	1	-1
376	0	0	1	-1	1	-1
377	0	0	-1	-1	-1	-1
378	0	0	-1	-1	-1	-1
379	0	0	-1	-1	-1	-1
380	0	0	-1	-1	-1	-1
381	0	0	-1	-1	-1	-1
382	0	0	-1	-1	-1	-1
383	0	0	-1	-1	-1	-1
384	0	0	1	-1	1	-1
385	0	0	1	-1	1	-1
386	0	0	-1	-1	-1	-1
387	0	0	-1	-1	-1	-1
388	0	0	-1	-1	-1	-1
389	0	0	1	-1	1	-1
390	0	0	1	-1	1	-1
391	0	0	-1	-1	-1	-1
392	0	0	-1	-1	-1	-1
393	0	0	-1	-1	-1	-1
394	0	0	-1	-1	-1	-1
395	0	0	1	-1	1	-1
396	0	0	1	-1	1	-1
397	0	0	1	-1	1	-1

373	-2	-1	0	-1	1	-2
374	-2	-1	0	-1	1	-2
375	-2	-1	0	-1	1	-2
376	-2	-1	0	-1	1	-2
377	-2	-1	0	-1	-1	-2
378	-2	-1	0	-1	-1	-2
379	-2	-1	0	-1	-1	-2
380	-2	-1	0	-1	-1	-2
381	-2	-1	0	-1	-1	-2
382	-2	-1	0	-1	-1	-2
383	-2	-1	0	-1	-1	-2
384	-2	-1	0	-1	1	-2
385	-2	-1	0	-1	1	-2
386	-2	-1	0	-1	-1	-2
387	-2	-1	0	-1	-1	-2
388	-2	-1	0	-1	-1	-2
389	-2	-1	0	-1	1	-2
390	-2	-1	0	-1	1	-2
391	-2	-1	0	-1	-1	-2
392	-2	-1	0	-1	-1	-2
393	-2	-1	0	-1	-1	-2
394	-2	-1	0	-1	-1	-2
395	-2	-1	0	-1	1	-2
396	-2	-1	0	-1	1	-2
397	-2	-1	0	-1	1	-2

398	0	0	1	-1	1	-1
399	0	0	1	-1	1	-1
400	0	0	1	-1	1	-1
401	0	0	1	-1	1	-1
402	0	0	1	-1	1	-1
403	0	0	1	-1	1	-1
404	0	0	1	-1	1	-1
405	0	0	-1	-1	-1	-1
406	0	0	1	-1	1	-1
407	0	0	-1	-1	-1	-1
408	0	0	1	-1	1	-1
409	0	0	1	-1	1	-1
410	0	0	-1	-1	-1	-1
411	0	0	-1	-1	-1	-1
412	0	0	-1	-1	-1	-1
413	0	0	-1	-1	-1	-1
414	0	0	1	-1	1	-1
415	0	0	1	-1	1	-1
416	0	0	1	-1	1	-1
417	0	0	1	-1	1	-1
418	0	0	-1	-1	-1	-1
419	0	0	1	-1	1	-1
420	0	0	1	-1	1	-1
421	0	0	1	-1	1	-1
422	0	0	1	-1	1	-1

398	-2	-1	0	-1	1	-2
399	-2	-1	0	-1	1	-2
400	-2	-1	0	-1	1	-2
401	-2	-1	0	-1	1	-2
402	-2	-1	0	-1	1	-2
403	-2	-1	0	-1	1	-2
404	-2	-1	0	-1	1	-2
405	-2	-1	0	-1	-1	-2
406	-2	-1	0	-1	1	-2
407	-2	-1	0	-1	-1	-2
408	-2	-1	0	-1	1	-2
409	-2	-1	0	-1	1	-2
410	-2	-1	0	-1	-1	-2
411	-2	-1	0	-1	-1	-2
412	-2	-1	0	-1	-1	-2
413	-2	-1	0	-1	-1	-2
414	-2	-1	0	-1	1	-2
415	-2	-1	0	-1	1	-2
416	-2	-1	0	-1	1	-2
417	-2	-1	0	-1	1	-2
418	-2	-1	0	-1	-1	-2
419	-2	-1	0	-1	1	-2
420	-2	-1	0	-1	1	-2
421	-2	-1	0	-1	1	-2
422	-2	-1	0	-1	1	-2

423	0	0	1	-1	1	-1
424	0	0	1	-1	1	-1
425	0	0	1	-1	1	-1
426	0	0	1	-1	1	-1
427	0	0	-1	-1	-1	-1
428	0	0	1	-1	1	-1
429	0	0	-1	-1	-1	-1
430	0	0	-1	-1	-1	-1
431	0	0	-1	-1	-1	-1
432	0	0	-1	-1	-1	-1
433	0	0	1	-1	1	-1
434	0	0	1	-1	1	-1
435	0	0	-1	-1	-1	-1
436	0	0	-1	-1	-1	-1
437	0	0	-1	-1	-1	-1
438	0	0	1	-1	1	-1
439	0	0	1	-1	1	-1
440	0	0	1	-1	1	-1
441	0	0	1	-1	1	-1
442	0	0	-1	-1	-1	-1
443	0	0	1	-1	1	-1
444	0	0	1	-1	1	-1
445	0	0	1	-1	1	-1
446	0	0	1	-1	1	-1
447	0	0	1	-1	1	-1

423	-2	-1	0	-1	1	-2
424	-2	-1	0	-1	1	-2
425	-2	-1	0	-1	1	-2
426	-2	-1	0	-1	1	-2
427	-2	-1	0	-1	-1	-2
428	-2	-1	0	-1	1	-2
429	-2	-1	0	-1	-1	-2
430	-2	-1	0	-1	-1	-2
431	-2	-1	0	-1	-1	-2
432	-2	-1	0	-1	-1	-2
433	-2	-1	0	-1	1	-2
434	-2	-1	0	-1	1	-2
435	-2	-1	0	-1	-1	-2
436	-2	-1	0	-1	-1	-2
437	-2	-1	0	-1	-1	-2
438	-2	-1	0	-1	1	-2
439	-2	-1	0	-1	1	-2
440	-2	-1	0	-1	1	-2
441	-2	-1	0	-1	1	-2
442	-2	-1	0	-1	-1	-2
443	-2	-1	0	-1	1	-2
444	-2	-1	0	-1	1	-2
445	-2	-1	0	-1	1	-2
446	-2	-1	0	-1	1	-2
447	-2	-1	0	-1	1	-2

448	0	0	1	-1	1	-1
449	0	0	1	-1	1	-1
450	0	0	1	-1	1	-1
451	0	0	1	-1	1	-1
452	0	0	-1	-1	-1	-1
453	0	0	1	-1	1	-1
454	0	0	-1	-1	-1	-1
455	0	0	-1	-1	-1	-1
456	0	0	-1	-1	-1	-1
457	0	0	1	-1	1	-1
458	0	0	1	-1	1	-1
459	0	0	-1	-1	-1	-1
460	0	0	-1	-1	-1	-1
461	0	0	-1	-1	-1	-1
462	0	0	-1	-1	-1	-1
463	0	0	1	-1	1	-1
464	0	0	1	-1	1	-1
465	0	0	1	-1	1	-1
466	0	0	-1	-1	-1	-1
467	0	0	1	-1	1	-1
468	0	0	1	-1	1	-1
469	0	0	1	-1	1	-1
470	0	0	1	-1	1	-1
471	0	0	1	-1	1	-1
472	0	0	1	-1	1	-1

448	-2	-1	0	-1	1	-2
449	-2	-1	0	-1	1	-2
450	-2	-1	0	-1	1	-2
451	-2	-1	0	-1	1	-2
452	-2	-1	0	-1	-1	-2
453	-2	-1	0	-1	1	-2
454	-2	-1	0	-1	-1	-2
455	-2	-1	0	-1	-1	-2
456	-2	-1	0	-1	-1	-2
457	-2	-1	0	-1	1	-2
458	-2	-1	0	-1	1	-2
459	-2	-1	0	-1	-1	-2
460	-2	-1	0	-1	-1	-2
461	-2	-1	0	-1	-1	-2
462	-2	-1	0	-1	-1	-2
463	-2	-1	0	-1	1	-2
464	-2	-1	0	-1	1	-2
465	-2	-1	0	-1	1	-2
466	-2	-1	0	-1	-1	-2
467	-2	-1	0	-1	1	-2
468	-2	-1	0	-1	1	-2
469	-2	-1	0	-1	1	-2
470	-2	-1	0	-1	1	-2
471	-2	-1	0	-1	1	-2
472	-2	-1	0	-1	1	-2

473	0	0	1	-1	1	-1
474	0	0	1	-1	1	-1
475	0	0	1	-1	1	-1
476	0	0	1	-1	1	-1
477	0	0	-1	-1	-1	-1
478	0	0	1	-1	1	-1
479	0	0	1	-1	1	-1
480	0	0	1	-1	1	-1
481	0	0	-1	-1	-1	-1
482	0	0	1	-1	1	-1
483	0	0	1	-1	1	-1
484	0	0	-1	-1	-1	-1
485	0	0	-1	-1	-1	-1
486	0	0	-1	-1	-1	-1
487	0	0	1	-1	1	-1
488	0	0	1	-1	1	-1
489	0	0	1	-1	1	-1
490	0	0	-1	-1	-1	-1
491	0	0	-1	-1	-1	-1
492	0	0	1	-1	1	-1
493	0	0	1	-1	1	-1
494	0	0	1	-1	1	-1
495	0	0	1	-1	1	-1
496	0	0	1	-1	1	-1
497	0	0	1	-1	1	-1

473	-2	-1	0	-1	1	-2
474	-2	-1	0	-1	1	-2
475	-2	-1	0	-1	1	-2
476	-2	-1	0	-1	1	-2
477	-2	-1	0	-1	-1	-2
478	-2	-1	0	-1	1	-2
479	-2	-1	0	-1	1	-2
480	-2	-1	0	-1	1	-2
481	-2	-1	0	-1	-1	-2
482	-2	-1	0	-1	1	-2
483	-2	-1	0	-1	1	-2
484	-2	-1	0	-1	-1	-2
485	-2	-1	0	-1	-1	-2
486	-2	-1	0	-1	-1	-2
487	-2	-1	0	-1	1	-2
488	-2	-1	0	-1	1	-2
489	-2	-1	0	-1	1	-2
490	-2	-1	0	-1	-1	-2
491	-2	-1	0	-1	-1	-2
492	-2	-1	0	-1	1	-2
493	-2	-1	0	-1	1	-2
494	-2	-1	0	-1	1	-2
495	-2	-1	0	-1	1	-2
496	-2	-1	0	-1	1	-2
497	-2	-1	0	-1	1	-2

498	0	0	1	-1	1	-1
499	0	0	1	-1	1	-1
500	0	0	1	-1	1	-1
501	0	0	1	-1	1	-1
502	0	0	1	-1	1	-1
503	0	0	1	-1	1	-1
504	0	0	-1	-1	-1	-1
505	0	0	-1	-1	-1	-1
506	0	0	-1	-1	-1	-1
507	0	0	1	-1	1	-1
508	0	0	1	-1	1	-1
509	0	0	-1	-1	-1	-1
510	0	0	-1	-1	-1	-1
511	0	0	1	-1	1	-1
512	0	0	1	-1	1	-1
513	0	0	1	-1	1	-1
514	0	0	-1	-1	-1	-1
515	0	0	-1	-1	-1	-1
516	0	0	-1	-1	-1	-1
517	0	0	-1	-1	-1	-1
518	0	0	-1	-1	-1	-1
519	0	0	1	-1	1	-1
520	0	0	1	-1	1	-1
521	0	0	1	-1	1	-1
522	0	0	1	-1	1	-1

498	-2	-1	0	-1	1	-2
499	-2	-1	0	-1	1	-2
500	-2	-1	0	-1	1	-2
501	-2	-1	0	-1	1	-2
502	-2	-1	0	-1	1	-2
503	-2	-1	0	-1	1	-2
504	-2	-1	0	-1	-1	-2
505	-2	-1	0	-1	-1	-2
506	-2	-1	0	-1	-1	-2
507	-2	-1	0	-1	1	-2
508	-2	-1	0	-1	1	-2
509	-2	-1	0	-1	-1	-2
510	-2	-1	0	-1	-1	-2
511	-2	-1	0	-1	1	-2
512	-2	-1	0	-1	1	-2
513	-2	-1	0	-1	1	-2
514	-2	-1	0	-1	-1	-2
515	-2	-1	0	-1	-1	-2
516	-2	-1	0	-1	-1	-2
517	-2	-1	0	-1	-1	-2
518	-2	-1	0	-1	-1	-2
519	-2	-1	0	-1	1	-2
520	-2	-1	0	-1	1	-2
521	-2	-1	0	-1	1	-2
522	-2	-1	0	-1	1	-2

523	0	0	1	-1	1	-1
524	0	0	1	-1	1	-1
525	0	0	1	-1	1	-1
526	0	0	1	-1	1	-1
527	0	0	1	-1	1	-1
528	0	0	1	-1	1	-1
529	0	0	1	-1	1	-1
530	0	0	-1	-1	-1	-1
531	0	0	1	-1	1	-1
532	0	0	1	-1	1	-1
533	0	0	-1	-1	-1	-1
534	0	0	-1	-1	-1	-1
535	0	0	1	-1	1	-1
536	0	0	-1	-1	-1	-1
537	0	0	-1	-1	-1	-1
538	0	0	-1	-1	-1	-1
539	0	0	-1	-1	-1	-1
540	0	0	-1	-1	-1	-1
541	0	0	1	-1	1	-1
542	0	0	1	-1	1	-1
543	0	0	1	-1	1	-1
544	0	0	1	-1	1	-1
545	0	0	1	-1	1	-1
546	0	0	1	-1	1	-1
547	0	0	1	-1	1	-1

523	-2	-1	0	-1	1	-2
524	-2	-1	0	-1	1	-2
525	-2	-1	0	-1	1	-2
526	-2	-1	0	-1	1	-2
527	-2	-1	0	-1	1	-2
528	-2	-1	0	-1	1	-2
529	-2	-1	0	-1	1	-2
530	-2	-1	0	-1	-1	-2
531	-2	-1	0	-1	1	-2
532	-2	-1	0	-1	1	-2
533	-2	-1	0	-1	-1	-2
534	-2	-1	0	-1	-1	-2
535	-2	-1	0	-1	1	-2
536	-2	-1	0	-1	-1	-2
537	-2	-1	0	-1	-1	-2
538	-2	-1	0	-1	-1	-2
539	-2	-1	0	-1	-1	-2
540	-2	-1	0	-1	-1	-2
541	-2	-1	0	-1	1	-2
542	-2	-1	0	-1	1	-2
543	-2	-1	0	-1	1	-2
544	-2	-1	0	-1	1	-2
545	-2	-1	0	-1	1	-2
546	-2	-1	0	-1	1	-2
547	-2	-1	0	-1	1	-2

548	0	0	1	-1	1	-1
549	0	0	1	-1	1	-1
550	0	0	1	-1	1	-1
551	0	0	1	-1	1	-1
552	0	0	1	-1	1	-1
553	0	0	1	-1	1	-1
554	0	0	1	-1	1	-1
555	0	0	1	-1	1	-1
556	0	0	1	-1	1	-1
557	0	0	1	-1	1	-1
558	0	0	1	-1	1	-1
559	0	0	1	-1	1	-1
560	0	0	1	-1	1	-1
561	0	0	1	-1	1	-1
562	0	0	1	-1	1	-1
563	0	0	1	-1	1	-1
564	0	0	1	-1	1	-1
565	0	0	1	-1	1	-1
566	0	0	1	-1	1	-1
567	0	0	1	-1	1	-1
568	0	0	1	-1	1	-1
569	0	0	1	-1	1	-1
570	0	0	1	-1	1	-1
571	0	0	1	-1	1	-1
572	0	0	1	-1	1	-1

548	-2	-1	0	-1	1	-2
549	-2	-1	0	-1	1	-2
550	-2	-1	0	-1	1	-2
551	-2	-1	0	-1	1	-2
552	-2	-1	0	-1	1	-2
553	-2	-1	0	-1	1	-2
554	-2	-1	0	-1	1	-2
555	-2	-1	0	-1	1	-2
556	-2	-1	0	-1	1	-2
557	-2	-1	0	-1	1	-2
558	-2	-1	0	-1	1	-2
559	-2	-1	0	-1	1	-2
560	-2	-1	0	-1	1	-2
561	-2	-1	0	-1	1	-2
562	-2	-1	0	-1	1	-2
563	-2	-1	0	-1	1	-2
564	-2	-1	0	-1	1	-2
565	-2	-1	0	-1	1	-2
566	-2	-1	0	-1	1	-2
567	-2	-1	0	-1	1	-2
568	-2	-1	0	-1	1	-2
569	-2	-1	0	-1	1	-2
570	-2	-1	0	-1	1	-2
571	-2	-1	0	-1	1	-2
572	-2	-1	0	-1	1	-2

573	0	0	1	-1	1	-1
574	0	0	1	-1	1	-1
575	0	0	1	-1	1	-1
576	0	0	1	-1	1	-1
577	0	0	1	-1	1	-1
578	0	0	1	-1	1	-1
579	0	0	1	-1	1	-1
580	0	0	1	-1	1	-1
581	0	0	1	-1	1	-1
582	0	0	1	-1	1	-1
583	0	0	1	-1	1	-1
584	0	0	1	-1	1	-1
585	0	0	1	-1	1	-1
586	0	0	1	-1	1	-1
587	0	0	1	-1	1	-1
588	0	0	1	-1	1	-1
589	0	0	1	-1	1	-1
590	0	0	1	-1	1	-1
591	0	0	1	-1	1	-1
592	0	0	1	-1	1	-1
593	0	0	1	-1	1	-1
594	0	0	1	-1	1	-1
595	0	0	1	-1	1	-1
596	0	0	1	-1	1	-1
597	0	0	1	-1	1	-1

573	-2	-1	0	-1	1	-2
574	-2	-1	0	-1	1	-2
575	-2	-1	0	-1	1	-2
576	-2	-1	0	-1	1	-2
577	-2	-1	0	-1	1	-2
578	-2	-1	0	-1	1	-2
579	-2	-1	0	-1	1	-2
580	-2	-1	0	-1	1	-2
581	-2	-1	0	-1	1	-2
582	-2	-1	0	-1	1	-2
583	-2	-1	0	-1	1	-2
584	-2	-1	0	-1	1	-2
585	-2	-1	0	-1	1	-2
586	-2	-1	0	-1	1	-2
587	-2	-1	0	-1	1	-2
588	-2	-1	0	-1	1	-2
589	-2	-1	0	-1	1	-2
590	-2	-1	0	-1	1	-2
591	-2	-1	0	-1	1	-2
592	-2	-1	0	-1	1	-2
593	-2	-1	0	-1	1	-2
594	-2	-1	0	-1	1	-2
595	-2	-1	0	-1	1	-2
596	-2	-1	0	-1	1	-2
597	-2	-1	0	-1	1	-2

598	0	0	1	-1	1	-1
599	0	0	1	-1	1	-1
600	0	0	1	-1	1	-1
601	0	0	1	-1	1	-1
602	0	0	1	-1	1	-1
603	0	0	1	-1	1	-1
604	0	0	1	-1	1	-1
605	0	0	1	-1	1	-1
606	0	0	1	-1	1	-1
607	0	0	1	-1	1	-1
608	0	0	1	-1	1	-1
609	0	0	1	-1	1	-1
610	0	0	1	-1	1	-1
611	0	0	1	-1	1	-1
612	0	0	1	-1	1	-1
613	0	0	1	-1	1	-1
614	0	0	1	-1	1	-1
615	0	0	1	-1	1	-1
616	0	0	1	-1	1	-1
617	0	0	1	-1	1	-1
618	0	0	1	-1	1	-1
619	0	0	1	-1	1	-1
620	0	0	1	-1	1	-1
621	0	0	1	-1	1	-1
622	0	0	1	-1	1	-1

598	-2	-1	0	-1	1	-2
599	-2	-1	0	-1	1	-2
600	-2	-1	0	-1	1	-2
601	-2	-1	0	-1	1	-2
602	-2	-1	0	-1	1	-2
603	-2	-1	0	-1	1	-2
604	-2	-1	0	-1	1	-2
605	-2	-1	0	-1	1	-2
606	-2	-1	0	-1	1	-2
607	-2	-1	0	-1	1	-2
608	-2	-1	0	-1	1	-2
609	-2	-1	0	-1	1	-2
610	-2	-1	0	-1	1	-2
611	-2	-1	0	-1	1	-2
612	-2	-1	0	-1	1	-2
613	-2	-1	0	-1	1	-2
614	-2	-1	0	-1	1	-2
615	-2	-1	0	-1	1	-2
616	-2	-1	0	-1	1	-2
617	-2	-1	0	-1	1	-2
618	-2	-1	0	-1	1	-2
619	-2	-1	0	-1	1	-2
620	-2	-1	0	-1	1	-2
621	-2	-1	0	-1	1	-2
622	-2	-1	0	-1	1	-2

623	0	0	1	-1	1	-1
624	0	0	1	-1	1	-1
625	0	0	1	-1	1	-1

623	-2	-1	0	-1	1	-2
624	-2	-1	0	-1	1	-2
625	-2	-1	0	-1	1	-2

KENDARAAN RODA DUA

NO	y_in(*net*)	y(*fnet*)	Δ BOBOT	Δ BIAS	ω_{baru}	b_{baru}
1	0	0	-1	1	-1	1
2	0	0	-1	1	-1	1
3	0	0	-1	1	-1	1
4	0	0	-1	1	-1	1
5	0	0	-1	1	-1	1
6	0	0	-1	1	-1	1
7	0	0	-1	1	-1	1
8	0	0	-1	1	-1	1
9	0	0	-1	1	-1	1
10	0	0	-1	1	-1	1
11	0	0	-1	1	-1	1
12	0	0	-1	1	-1	1
13	0	0	-1	1	-1	1
14	0	0	-1	1	-1	1
15	0	0	-1	1	-1	1
16	0	0	-1	1	-1	1
17	0	0	-1	1	-1	1

NO	y_in(*net*)	y(*fnet*)	Δ BOBOT	Δ BIAS	ω_{baru}	b_{baru}
1	2	1	0	1	-1	2
2	2	1	0	1	-1	2
3	2	1	0	1	-1	2
4	2	1	0	1	-1	2
5	2	1	0	1	-1	2
6	2	1	0	1	-1	2
7	2	1	0	1	-1	2
8	2	1	0	1	-1	2
9	2	1	0	1	-1	2
10	2	1	0	1	-1	2
11	2	1	0	1	-1	2
12	2	1	0	1	-1	2
13	2	1	0	1	-1	2
14	2	1	0	1	-1	2
15	2	1	0	1	-1	2
16	2	1	0	1	-1	2
17	2	1	0	1	-1	2

18	0	0	-1	1	-1	1
19	0	0	-1	1	-1	1
20	0	0	-1	1	-1	1
21	0	0	-1	1	-1	1
22	0	0	-1	1	-1	1
23	0	0	-1	1	-1	1
24	0	0	-1	1	-1	1
25	0	0	-1	1	-1	1
26	0	0	-1	1	-1	1
27	0	0	-1	1	-1	1
28	0	0	-1	1	-1	1
29	0	0	-1	1	-1	1
30	0	0	-1	1	-1	1
31	0	0	-1	1	-1	1
32	0	0	-1	1	-1	1
33	0	0	-1	1	-1	1
34	0	0	-1	1	-1	1
35	0	0	-1	1	-1	1
36	0	0	1	1	1	1
37	0	0	-1	1	-1	1
38	0	0	-1	1	-1	1
39	0	0	1	1	1	1
40	0	0	-1	1	-1	1
41	0	0	-1	1	-1	1
42	0	0	-1	1	-1	1

18	2	1	0	1	-1	2
19	2	1	0	1	-1	2
20	2	1	0	1	-1	2
21	2	1	0	1	-1	2
22	2	1	0	1	-1	2
23	2	1	0	1	-1	2
24	2	1	0	1	-1	2
25	2	1	0	1	-1	2
26	2	1	0	1	-1	2
27	2	1	0	1	-1	2
28	2	1	0	1	-1	2
29	2	1	0	1	-1	2
30	2	1	0	1	-1	2
31	2	1	0	1	-1	2
32	2	1	0	1	-1	2
33	2	1	0	1	-1	2
34	2	1	0	1	-1	2
35	2	1	0	1	-1	2
36	2	1	0	1	1	2
37	2	1	0	1	-1	2
38	2	1	0	1	-1	2
39	2	1	0	1	1	2
40	2	1	0	1	-1	2
41	2	1	0	1	-1	2
42	2	1	0	1	-1	2

43	0	0	-1	1	-1	1
44	0	0	-1	1	-1	1
45	0	0	-1	1	-1	1
46	0	0	-1	1	-1	1
47	0	0	-1	1	-1	1
48	0	0	-1	1	-1	1
49	0	0	-1	1	-1	1
50	0	0	-1	1	-1	1
51	0	0	-1	1	-1	1
52	0	0	-1	1	-1	1
53	0	0	-1	1	-1	1
54	0	0	-1	1	-1	1
55	0	0	-1	1	-1	1
56	0	0	-1	1	-1	1
57	0	0	1	1	1	1
58	0	0	-1	1	-1	1
59	0	0	-1	1	-1	1
60	0	0	1	1	1	1
61	0	0	1	1	1	1
62	0	0	-1	1	-1	1
63	0	0	1	1	1	1
64	0	0	-1	1	-1	1
65	0	0	-1	1	-1	1
66	0	0	1	1	1	1
67	0	0	1	1	1	1

43	2	1	0	1	-1	2
44	2	1	0	1	-1	2
45	2	1	0	1	-1	2
46	2	1	0	1	-1	2
47	2	1	0	1	-1	2
48	2	1	0	1	-1	2
49	2	1	0	1	-1	2
50	2	1	0	1	-1	2
51	2	1	0	1	-1	2
52	2	1	0	1	-1	2
53	2	1	0	1	-1	2
54	2	1	0	1	-1	2
55	2	1	0	1	-1	2
56	2	1	0	1	-1	2
57	2	1	0	1	1	2
58	2	1	0	1	-1	2
59	2	1	0	1	-1	2
60	2	1	0	1	1	2
61	2	1	0	1	1	2
62	2	1	0	1	-1	2
63	2	1	0	1	1	2
64	2	1	0	1	-1	2
65	2	1	0	1	-1	2
66	2	1	0	1	1	2
67	2	1	0	1	1	2

68	0	0	-1	1	-1	1
69	0	0	-1	1	-1	1
70	0	0	-1	1	-1	1
71	0	0	-1	1	-1	1
72	0	0	-1	1	-1	1
73	0	0	-1	1	-1	1
74	0	0	-1	1	-1	1
75	0	0	-1	1	-1	1
76	0	0	-1	1	-1	1
77	0	0	-1	1	-1	1
78	0	0	-1	1	-1	1
79	0	0	-1	1	-1	1
80	0	0	-1	1	-1	1
81	0	0	-1	1	-1	1
82	0	0	1	1	1	1
83	0	0	1	1	1	1
84	0	0	-1	1	-1	1
85	0	0	1	1	1	1
86	0	0	-1	1	-1	1
87	0	0	1	1	1	1
88	0	0	-1	1	-1	1
89	0	0	-1	1	-1	1
90	0	0	1	1	1	1
91	0	0	1	1	1	1
92	0	0	-1	1	-1	1

68	2	1	0	1	-1	2
69	2	1	0	1	-1	2
70	2	1	0	1	-1	2
71	2	1	0	1	-1	2
72	2	1	0	1	-1	2
73	2	1	0	1	-1	2
74	2	1	0	1	-1	2
75	2	1	0	1	-1	2
76	2	1	0	1	-1	2
77	2	1	0	1	-1	2
78	2	1	0	1	-1	2
79	2	1	0	1	-1	2
80	2	1	0	1	-1	2
81	2	1	0	1	-1	2
82	2	1	0	1	1	2
83	2	1	0	1	1	2
84	2	1	0	1	-1	2
85	2	1	0	1	1	2
86	2	1	0	1	-1	2
87	2	1	0	1	1	2
88	2	1	0	1	-1	2
89	2	1	0	1	-1	2
90	2	1	0	1	1	2
91	2	1	0	1	1	2
92	2	1	0	1	-1	2

93	0	0	-1	1	-1	1
94	0	0	1	1	1	1
95	0	0	-1	1	-1	1
96	0	0	-1	1	-1	1
97	0	0	-1	1	-1	1
98	0	0	-1	1	-1	1
99	0	0	-1	1	-1	1
100	0	0	-1	1	-1	1
101	0	0	-1	1	-1	1
102	0	0	-1	1	-1	1
103	0	0	-1	1	-1	1
104	0	0	-1	1	-1	1
105	0	0	-1	1	-1	1
106	0	0	-1	1	-1	1
107	0	0	1	1	1	1
108	0	0	1	1	1	1
109	0	0	1	1	1	1
110	0	0	-1	1	-1	1
111	0	0	1	1	1	1
112	0	0	-1	1	-1	1
113	0	0	-1	1	-1	1
114	0	0	1	1	1	1
115	0	0	1	1	1	1
116	0	0	1	1	1	1
117	0	0	1	1	1	1

93	2	1	0	1	-1	2
94	2	1	0	1	1	2
95	2	1	0	1	-1	2
96	2	1	0	1	-1	2
97	2	1	0	1	-1	2
98	2	1	0	1	-1	2
99	2	1	0	1	-1	2
100	2	1	0	1	-1	2
101	2	1	0	1	-1	2
102	2	1	0	1	-1	2
103	2	1	0	1	-1	2
104	2	1	0	1	-1	2
105	2	1	0	1	-1	2
106	2	1	0	1	-1	2
107	2	1	0	1	1	2
108	2	1	0	1	1	2
109	2	1	0	1	1	2
110	2	1	0	1	-1	2
111	2	1	0	1	1	2
112	2	1	0	1	-1	2
113	2	1	0	1	-1	2
114	2	1	0	1	1	2
115	2	1	0	1	1	2
116	2	1	0	1	1	2
117	2	1	0	1	1	2

118	0	0	-1	1	-1	1
119	0	0	1	1	1	1
120	0	0	-1	1	-1	1
121	0	0	-1	1	-1	1
122	0	0	-1	1	-1	1
123	0	0	-1	1	-1	1
124	0	0	-1	1	-1	1
125	0	0	-1	1	-1	1
126	0	0	-1	1	-1	1
127	0	0	-1	1	-1	1
128	0	0	-1	1	-1	1
129	0	0	-1	1	-1	1
130	0	0	-1	1	-1	1
131	0	0	-1	1	-1	1
132	0	0	1	1	1	1
133	0	0	1	1	1	1
134	0	0	1	1	1	1
135	0	0	1	1	1	1
136	0	0	-1	1	-1	1
137	0	0	1	1	1	1
138	0	0	1	1	1	1
139	0	0	1	1	1	1
140	0	0	-1	1	-1	1
141	0	0	1	1	1	1
142	0	0	1	1	1	1

118	2	1	0	1	-1	2
119	2	1	0	1	1	2
120	2	1	0	1	-1	2
121	2	1	0	1	-1	2
122	2	1	0	1	-1	2
123	2	1	0	1	-1	2
124	2	1	0	1	-1	2
125	2	1	0	1	-1	2
126	2	1	0	1	-1	2
127	2	1	0	1	-1	2
128	2	1	0	1	-1	2
129	2	1	0	1	-1	2
130	2	1	0	1	-1	2
131	2	1	0	1	-1	2
132	2	1	0	1	1	2
133	2	1	0	1	1	2
134	2	1	0	1	1	2
135	2	1	0	1	1	2
136	2	1	0	1	-1	2
137	2	1	0	1	1	2
138	2	1	0	1	1	2
139	2	1	0	1	1	2
140	2	1	0	1	-1	2
141	2	1	0	1	1	2
142	2	1	0	1	1	2

143	0	0	-1	1	-1	1
144	0	0	1	1	1	1
145	0	0	-1	1	-1	1
146	0	0	-1	1	-1	1
147	0	0	-1	1	-1	1
148	0	0	-1	1	-1	1
149	0	0	-1	1	-1	1
150	0	0	-1	1	-1	1
151	0	0	-1	1	-1	1
152	0	0	-1	1	-1	1
153	0	0	-1	1	-1	1
154	0	0	-1	1	-1	1
155	0	0	-1	1	-1	1
156	0	0	-1	1	-1	1
157	0	0	-1	1	-1	1
158	0	0	1	1	1	1
159	0	0	1	1	1	1
160	0	0	1	1	1	1
161	0	0	-1	1	-1	1
162	0	0	1	1	1	1
163	0	0	1	1	1	1
164	0	0	1	1	1	1
165	0	0	1	1	1	1
166	0	0	1	1	1	1
167	0	0	1	1	1	1

143	2	1	0	1	-1	2
144	2	1	0	1	1	2
145	2	1	0	1	-1	2
146	2	1	0	1	-1	2
147	2	1	0	1	-1	2
148	2	1	0	1	-1	2
149	2	1	0	1	-1	2
150	2	1	0	1	-1	2
151	2	1	0	1	-1	2
152	2	1	0	1	-1	2
153	2	1	0	1	-1	2
154	2	1	0	1	-1	2
155	2	1	0	1	-1	2
156	2	1	0	1	-1	2
157	2	1	0	1	-1	2
158	2	1	0	1	1	2
159	2	1	0	1	1	2
160	2	1	0	1	1	2
161	2	1	0	1	-1	2
162	2	1	0	1	1	2
163	2	1	0	1	1	2
164	2	1	0	1	1	2
165	2	1	0	1	1	2
166	2	1	0	1	1	2
167	2	1	0	1	1	2

168	0	0	1	1	1	1
169	0	0	1	1	1	1
170	0	0	-1	1	-1	1
171	0	0	-1	1	-1	1
172	0	0	-1	1	-1	1
173	0	0	-1	1	-1	1
174	0	0	-1	1	-1	1
175	0	0	-1	1	-1	1
176	0	0	-1	1	-1	1
177	0	0	-1	1	-1	1
178	0	0	-1	1	-1	1
179	0	0	-1	1	-1	1
180	0	0	-1	1	-1	1
181	0	0	-1	1	-1	1
182	0	0	-1	1	-1	1
183	0	0	1	1	1	1
184	0	0	1	1	1	1
185	0	0	1	1	1	1
186	0	0	1	1	1	1
187	0	0	-1	1	-1	1
188	0	0	1	1	1	1
189	0	0	1	1	1	1
190	0	0	1	1	1	1
191	0	0	-1	1	-1	1
192	0	0	1	1	1	1

168	2	1	0	1	1	2
169	2	1	0	1	1	2
170	2	1	0	1	-1	2
171	2	1	0	1	-1	2
172	2	1	0	1	-1	2
173	2	1	0	1	-1	2
174	2	1	0	1	-1	2
175	2	1	0	1	-1	2
176	2	1	0	1	-1	2
177	2	1	0	1	-1	2
178	2	1	0	1	-1	2
179	2	1	0	1	-1	2
180	2	1	0	1	-1	2
181	2	1	0	1	-1	2
182	2	1	0	1	-1	2
183	2	1	0	1	1	2
184	2	1	0	1	1	2
185	2	1	0	1	1	2
186	2	1	0	1	1	2
187	2	1	0	1	-1	2
188	2	1	0	1	1	2
189	2	1	0	1	1	2
190	2	1	0	1	1	2
191	2	1	0	1	-1	2
192	2	1	0	1	1	2

193	0	0	1	1	1	1
194	0	0	-1	1	-1	1
195	0	0	-1	1	-1	1
196	0	0	-1	1	-1	1
197	0	0	-1	1	-1	1
198	0	0	-1	1	-1	1
199	0	0	-1	1	-1	1
200	0	0	-1	1	-1	1
201	0	0	-1	1	-1	1
202	0	0	1	1	1	1
203	0	0	-1	1	-1	1
204	0	0	-1	1	-1	1
205	0	0	-1	1	-1	1
206	0	0	-1	1	-1	1
207	0	0	-1	1	-1	1
208	0	0	1	1	1	1
209	0	0	1	1	1	1
210	0	0	1	1	1	1
211	0	0	1	1	1	1
212	0	0	-1	1	-1	1
213	0	0	1	1	1	1
214	0	0	1	1	1	1
215	0	0	1	1	1	1
216	0	0	1	1	1	1
217	0	0	1	1	1	1

193	2	1	0	1	1	2
194	2	1	0	1	-1	2
195	2	1	0	1	-1	2
196	2	1	0	1	-1	2
197	2	1	0	1	-1	2
198	2	1	0	1	-1	2
199	2	1	0	1	-1	2
200	2	1	0	1	-1	2
201	2	1	0	1	-1	2
202	2	1	0	1	1	2
203	2	1	0	1	-1	2
204	2	1	0	1	-1	2
205	2	1	0	1	-1	2
206	2	1	0	1	-1	2
207	2	1	0	1	-1	2
208	2	1	0	1	1	2
209	2	1	0	1	1	2
210	2	1	0	1	1	2
211	2	1	0	1	1	2
212	2	1	0	1	-1	2
213	2	1	0	1	1	2
214	2	1	0	1	1	2
215	2	1	0	1	1	2
216	2	1	0	1	1	2
217	2	1	0	1	1	2

218	0	0	1	1	1	1
219	0	0	-1	1	-1	1
220	0	0	-1	1	-1	1
221	0	0	-1	1	-1	1
222	0	0	-1	1	-1	1
223	0	0	-1	1	-1	1
224	0	0	-1	1	-1	1
225	0	0	-1	1	-1	1
226	0	0	-1	1	-1	1
227	0	0	1	1	1	1
228	0	0	1	1	1	1
229	0	0	-1	1	-1	1
230	0	0	-1	1	-1	1
231	0	0	-1	1	-1	1
232	0	0	-1	1	-1	1
233	0	0	-1	1	-1	1
234	0	0	1	1	1	1
235	0	0	1	1	1	1
236	0	0	1	1	1	1
237	0	0	1	1	1	1
238	0	0	-1	1	-1	1
239	0	0	1	1	1	1
240	0	0	1	1	1	1
241	0	0	-1	1	-1	1
242	0	0	1	1	1	1

218	2	1	0	1	1	2
219	2	1	0	1	-1	2
220	2	1	0	1	-1	2
221	2	1	0	1	-1	2
222	2	1	0	1	-1	2
223	2	1	0	1	-1	2
224	2	1	0	1	-1	2
225	2	1	0	1	-1	2
226	2	1	0	1	-1	2
227	2	1	0	1	1	2
228	2	1	0	1	1	2
229	2	1	0	1	-1	2
230	2	1	0	1	-1	2
231	2	1	0	1	-1	2
232	2	1	0	1	-1	2
233	2	1	0	1	-1	2
234	2	1	0	1	1	2
235	2	1	0	1	1	2
236	2	1	0	1	1	2
237	2	1	0	1	1	2
238	2	1	0	1	-1	2
239	2	1	0	1	1	2
240	2	1	0	1	1	2
241	2	1	0	1	-1	2
242	2	1	0	1	1	2

243	0	0	1	1	1	1
244	0	0	-1	1	-1	1
245	0	0	-1	1	-1	1
246	0	0	-1	1	-1	1
247	0	0	-1	1	-1	1
248	0	0	-1	1	-1	1
249	0	0	-1	1	-1	1
250	0	0	-1	1	-1	1
251	0	0	-1	1	-1	1
252	0	0	-1	1	-1	1
253	0	0	-1	1	-1	1
254	0	0	-1	1	-1	1
255	0	0	-1	1	-1	1
256	0	0	-1	1	-1	1
257	0	0	-1	1	-1	1
258	0	0	-1	1	-1	1
259	0	0	1	1	1	1
260	0	0	1	1	1	1
261	0	0	1	1	1	1
262	0	0	1	1	1	1
263	0	0	-1	1	-1	1
264	0	0	1	1	1	1
265	0	0	1	1	1	1
266	0	0	-1	1	-1	1
267	0	0	1	1	1	1

243	2	1	0	1	1	2
244	2	1	0	1	-1	2
245	2	1	0	1	-1	2
246	2	1	0	1	-1	2
247	2	1	0	1	-1	2
248	2	1	0	1	-1	2
249	2	1	0	1	-1	2
250	2	1	0	1	-1	2
251	2	1	0	1	-1	2
252	2	1	0	1	-1	2
253	2	1	0	1	-1	2
254	2	1	0	1	-1	2
255	2	1	0	1	-1	2
256	2	1	0	1	-1	2
257	2	1	0	1	-1	2
258	2	1	0	1	-1	2
259	2	1	0	1	1	2
260	2	1	0	1	1	2
261	2	1	0	1	1	2
262	2	1	0	1	1	2
263	2	1	0	1	-1	2
264	2	1	0	1	1	2
265	2	1	0	1	1	2
266	2	1	0	1	-1	2
267	2	1	0	1	1	2

268	0	0	1	1	1	1
269	0	0	-1	1	-1	1
270	0	0	-1	1	-1	1
271	0	0	-1	1	-1	1
272	0	0	-1	1	-1	1
273	0	0	-1	1	-1	1
274	0	0	-1	1	-1	1
275	0	0	-1	1	-1	1
276	0	0	-1	1	-1	1
277	0	0	-1	1	-1	1
278	0	0	-1	1	-1	1
279	0	0	-1	1	-1	1
280	0	0	1	1	1	1
281	0	0	-1	1	-1	1
282	0	0	-1	1	-1	1
283	0	0	-1	1	-1	1
284	0	0	-1	1	-1	1
285	0	0	1	1	1	1
286	0	0	-1	1	-1	1
287	0	0	-1	1	-1	1
288	0	0	-1	1	-1	1
289	0	0	-1	1	-1	1
290	0	0	-1	1	-1	1
291	0	0	1	1	1	1
292	0	0	1	1	1	1

268	2	1	0	1	1	2
269	2	1	0	1	-1	2
270	2	1	0	1	-1	2
271	2	1	0	1	-1	2
272	2	1	0	1	-1	2
273	2	1	0	1	-1	2
274	2	1	0	1	-1	2
275	2	1	0	1	-1	2
276	2	1	0	1	-1	2
277	2	1	0	1	-1	2
278	2	1	0	1	-1	2
279	2	1	0	1	-1	2
280	2	1	0	1	1	2
281	2	1	0	1	-1	2
282	2	1	0	1	-1	2
283	2	1	0	1	-1	2
284	2	1	0	1	-1	2
285	2	1	0	1	1	2
286	2	1	0	1	-1	2
287	2	1	0	1	-1	2
288	2	1	0	1	-1	2
289	2	1	0	1	-1	2
290	2	1	0	1	-1	2
291	2	1	0	1	1	2
292	2	1	0	1	1	2

293	0	0	1	1	1	1
294	0	0	-1	1	-1	1
295	0	0	-1	1	-1	1
296	0	0	-1	1	-1	1
297	0	0	-1	1	-1	1
298	0	0	-1	1	-1	1
299	0	0	-1	1	-1	1
300	0	0	-1	1	-1	1
301	0	0	-1	1	-1	1
302	0	0	-1	1	-1	1
303	0	0	-1	1	-1	1
304	0	0	-1	1	-1	1
305	0	0	1	1	1	1
306	0	0	1	1	1	1
307	0	0	-1	1	-1	1
308	0	0	-1	1	-1	1
309	0	0	-1	1	-1	1
310	0	0	-1	1	-1	1
311	0	0	-1	1	-1	1
312	0	0	-1	1	-1	1
313	0	0	1	1	1	1
314	0	0	1	1	1	1
315	0	0	1	1	1	1
316	0	0	-1	1	-1	1
317	0	0	-1	1	-1	1

293	2	1	0	1	1	2
294	2	1	0	1	-1	2
295	2	1	0	1	-1	2
296	2	1	0	1	-1	2
297	2	1	0	1	-1	2
298	2	1	0	1	-1	2
299	2	1	0	1	-1	2
300	2	1	0	1	-1	2
301	2	1	0	1	-1	2
302	2	1	0	1	-1	2
303	2	1	0	1	-1	2
304	2	1	0	1	-1	2
305	2	1	0	1	1	2
306	2	1	0	1	1	2
307	2	1	0	1	-1	2
308	2	1	0	1	-1	2
309	2	1	0	1	-1	2
310	2	1	0	1	-1	2
311	2	1	0	1	-1	2
312	2	1	0	1	-1	2
313	2	1	0	1	1	2
314	2	1	0	1	1	2
315	2	1	0	1	1	2
316	2	1	0	1	-1	2
317	2	1	0	1	-1	2

318	0	0	1	1	1	1
319	0	0	-1	1	-1	1
320	0	0	-1	1	-1	1
321	0	0	-1	1	-1	1
322	0	0	-1	1	-1	1
323	0	0	-1	1	-1	1
324	0	0	-1	1	-1	1
325	0	0	-1	1	-1	1
326	0	0	-1	1	-1	1
327	0	0	-1	1	-1	1
328	0	0	-1	1	-1	1
329	0	0	-1	1	-1	1
330	0	0	1	1	1	1
331	0	0	1	1	1	1
332	0	0	-1	1	-1	1
333	0	0	-1	1	-1	1
334	0	0	-1	1	-1	1
335	0	0	-1	1	-1	1
336	0	0	-1	1	-1	1
337	0	0	-1	1	-1	1
338	0	0	1	1	1	1
339	0	0	1	1	1	1
340	0	0	1	1	1	1
341	0	0	1	1	1	1
342	0	0	1	1	1	1

318	2	1	0	1	1	2
319	2	1	0	1	-1	2
320	2	1	0	1	-1	2
321	2	1	0	1	-1	2
322	2	1	0	1	-1	2
323	2	1	0	1	-1	2
324	2	1	0	1	-1	2
325	2	1	0	1	-1	2
326	2	1	0	1	-1	2
327	2	1	0	1	-1	2
328	2	1	0	1	-1	2
329	2	1	0	1	-1	2
330	2	1	0	1	1	2
331	2	1	0	1	1	2
332	2	1	0	1	-1	2
333	2	1	0	1	-1	2
334	2	1	0	1	-1	2
335	2	1	0	1	-1	2
336	2	1	0	1	-1	2
337	2	1	0	1	-1	2
338	2	1	0	1	1	2
339	2	1	0	1	1	2
340	2	1	0	1	1	2
341	2	1	0	1	1	2
342	2	1	0	1	1	2

343	0	0	-1	1	-1	1
344	0	0	-1	1	-1	1
345	0	0	-1	1	-1	1
346	0	0	-1	1	-1	1
347	0	0	-1	1	-1	1
348	0	0	-1	1	-1	1
349	0	0	-1	1	-1	1
350	0	0	-1	1	-1	1
351	0	0	-1	1	-1	1
352	0	0	-1	1	-1	1
353	0	0	-1	1	-1	1
354	0	0	-1	1	-1	1
355	0	0	-1	1	-1	1
356	0	0	1	1	1	1
357	0	0	1	1	1	1
358	0	0	-1	1	-1	1
359	0	0	-1	1	-1	1
360	0	0	-1	1	-1	1
361	0	0	-1	1	-1	1
362	0	0	1	1	1	1
363	0	0	1	1	1	1
364	0	0	1	1	1	1
365	0	0	1	1	1	1
366	0	0	1	1	1	1
367	0	0	1	1	1	1

343	2	1	0	1	-1	2
344	2	1	0	1	-1	2
345	2	1	0	1	-1	2
346	2	1	0	1	-1	2
347	2	1	0	1	-1	2
348	2	1	0	1	-1	2
349	2	1	0	1	-1	2
350	2	1	0	1	-1	2
351	2	1	0	1	-1	2
352	2	1	0	1	-1	2
353	2	1	0	1	-1	2
354	2	1	0	1	-1	2
355	2	1	0	1	-1	2
356	2	1	0	1	1	2
357	2	1	0	1	1	2
358	2	1	0	1	-1	2
359	2	1	0	1	-1	2
360	2	1	0	1	-1	2
361	2	1	0	1	-1	2
362	2	1	0	1	1	2
363	2	1	0	1	1	2
364	2	1	0	1	1	2
365	2	1	0	1	1	2
366	2	1	0	1	1	2
367	2	1	0	1	1	2

368	0	0	1	1	1	1
369	0	0	-1	1	-1	1
370	0	0	-1	1	-1	1
371	0	0	-1	1	-1	1
372	0	0	-1	1	-1	1
373	0	0	-1	1	-1	1
374	0	0	-1	1	-1	1
375	0	0	-1	1	-1	1
376	0	0	-1	1	-1	1
377	0	0	-1	1	-1	1
378	0	0	-1	1	-1	1
379	0	0	1	1	1	1
380	0	0	-1	1	-1	1
381	0	0	1	1	1	1
382	0	0	1	1	1	1
383	0	0	-1	1	-1	1
384	0	0	-1	1	-1	1
385	0	0	-1	1	-1	1
386	0	0	-1	1	-1	1
387	0	0	1	1	1	1
388	0	0	1	1	1	1
389	0	0	1	1	1	1
390	0	0	1	1	1	1
391	0	0	1	1	1	1
392	0	0	1	1	1	1

368	2	1	0	1	1	2
369	2	1	0	1	-1	2
370	2	1	0	1	-1	2
371	2	1	0	1	-1	2
372	2	1	0	1	-1	2
373	2	1	0	1	-1	2
374	2	1	0	1	-1	2
375	2	1	0	1	-1	2
376	2	1	0	1	-1	2
377	2	1	0	1	-1	2
378	2	1	0	1	-1	2
379	2	1	0	1	1	2
380	2	1	0	1	-1	2
381	2	1	0	1	1	2
382	2	1	0	1	1	2
383	2	1	0	1	-1	2
384	2	1	0	1	-1	2
385	2	1	0	1	-1	2
386	2	1	0	1	-1	2
387	2	1	0	1	1	2
388	2	1	0	1	1	2
389	2	1	0	1	1	2
390	2	1	0	1	1	2
391	2	1	0	1	1	2
392	2	1	0	1	1	2

393	0	0	1	1	1	1
394	0	0	-1	1	-1	1
395	0	0	-1	1	-1	1
396	0	0	-1	1	-1	1
397	0	0	-1	1	-1	1
398	0	0	-1	1	-1	1
399	0	0	-1	1	-1	1
400	0	0	-1	1	-1	1
401	0	0	-1	1	-1	1
402	0	0	-1	1	-1	1
403	0	0	-1	1	-1	1
404	0	0	1	1	1	1
405	0	0	1	1	1	1
406	0	0	-1	1	-1	1
407	0	0	1	1	1	1
408	0	0	-1	1	-1	1
409	0	0	-1	1	-1	1
410	0	0	-1	1	-1	1
411	0	0	1	1	1	1
412	0	0	1	1	1	1
413	0	0	1	1	1	1
414	0	0	1	1	1	1
415	0	0	1	1	1	1
416	0	0	1	1	1	1
417	0	0	1	1	1	1

393	2	1	0	1	1	2
394	2	1	0	1	-1	2
395	2	1	0	1	-1	2
396	2	1	0	1	-1	2
397	2	1	0	1	-1	2
398	2	1	0	1	-1	2
399	2	1	0	1	-1	2
400	2	1	0	1	-1	2
401	2	1	0	1	-1	2
402	2	1	0	1	-1	2
403	2	1	0	1	-1	2
404	2	1	0	1	1	2
405	2	1	0	1	1	2
406	2	1	0	1	-1	2
407	2	1	0	1	1	2
408	2	1	0	1	-1	2
409	2	1	0	1	-1	2
410	2	1	0	1	-1	2
411	2	1	0	1	1	2
412	2	1	0	1	1	2
413	2	1	0	1	1	2
414	2	1	0	1	1	2
415	2	1	0	1	1	2
416	2	1	0	1	1	2
417	2	1	0	1	1	2

418	0	0	1	1	1	1
419	0	0	1	1	1	1
420	0	0	-1	1	-1	1
421	0	0	-1	1	-1	1
422	0	0	-1	1	-1	1
423	0	0	-1	1	-1	1
424	0	0	-1	1	-1	1
425	0	0	-1	1	-1	1
426	0	0	-1	1	-1	1
427	0	0	-1	1	-1	1
428	0	0	-1	1	-1	1
429	0	0	-1	1	-1	1
430	0	0	1	1	1	1
431	0	0	1	1	1	1
432	0	0	-1	1	-1	1
433	0	0	-1	1	-1	1
434	0	0	-1	1	-1	1
435	0	0	-1	1	-1	1
436	0	0	1	1	1	1
437	0	0	1	1	1	1
438	0	0	1	1	1	1
439	0	0	1	1	1	1
440	0	0	1	1	1	1
441	0	0	1	1	1	1
442	0	0	1	1	1	1

418	2	1	0	1	1	2
419	2	1	0	1	1	2
420	2	1	0	1	-1	2
421	2	1	0	1	-1	2
422	2	1	0	1	-1	2
423	2	1	0	1	-1	2
424	2	1	0	1	-1	2
425	2	1	0	1	-1	2
426	2	1	0	1	-1	2
427	2	1	0	1	-1	2
428	2	1	0	1	-1	2
429	2	1	0	1	-1	2
430	2	1	0	1	1	2
431	2	1	0	1	1	2
432	2	1	0	1	-1	2
433	2	1	0	1	-1	2
434	2	1	0	1	-1	2
435	2	1	0	1	-1	2
436	2	1	0	1	1	2
437	2	1	0	1	1	2
438	2	1	0	1	1	2
439	2	1	0	1	1	2
440	2	1	0	1	1	2
441	2	1	0	1	1	2
442	2	1	0	1	1	2

443	0	0	-1	1	-1	1
444	0	0	-1	1	-1	1
445	0	0	1	1	1	1
446	0	0	1	1	1	1
447	0	0	-1	1	-1	1
448	0	0	-1	1	-1	1
449	0	0	-1	1	-1	1
450	0	0	-1	1	-1	1
451	0	0	-1	1	-1	1
452	0	0	1	1	1	1
453	0	0	-1	1	-1	1
454	0	0	-1	1	-1	1
455	0	0	-1	1	-1	1
456	0	0	-1	1	-1	1
457	0	0	-1	1	-1	1
458	0	0	-1	1	-1	1
459	0	0	-1	1	-1	1
460	0	0	-1	1	-1	1
461	0	0	1	1	1	1
462	0	0	1	1	1	1
463	0	0	1	1	1	1
464	0	0	1	1	1	1
465	0	0	-1	1	-1	1
466	0	0	-1	1	-1	1
467	0	0	1	1	1	1

443	2	1	0	1	-1	2
444	2	1	0	1	-1	2
445	2	1	0	1	1	2
446	2	1	0	1	1	2
447	2	1	0	1	-1	2
448	2	1	0	1	-1	2
449	2	1	0	1	-1	2
450	2	1	0	1	-1	2
451	2	1	0	1	-1	2
452	2	1	0	1	1	2
453	2	1	0	1	-1	2
454	2	1	0	1	-1	2
455	2	1	0	1	-1	2
456	2	1	0	1	-1	2
457	2	1	0	1	-1	2
458	2	1	0	1	-1	2
459	2	1	0	1	-1	2
460	2	1	0	1	-1	2
461	2	1	0	1	1	2
462	2	1	0	1	1	2
463	2	1	0	1	1	2
464	2	1	0	1	1	2
465	2	1	0	1	-1	2
466	2	1	0	1	-1	2
467	2	1	0	1	1	2

468	0	0	1	1	1	1
469	0	0	-1	1	-1	1
470	0	0	-1	1	-1	1
471	0	0	-1	1	-1	1
472	0	0	1	1	1	1
473	0	0	-1	1	-1	1
474	0	0	-1	1	-1	1
475	0	0	-1	1	-1	1
476	0	0	-1	1	-1	1
477	0	0	-1	1	-1	1
478	0	0	-1	1	-1	1
479	0	0	-1	1	-1	1
480	0	0	-1	1	-1	1
481	0	0	-1	1	-1	1
482	0	0	-1	1	-1	1
483	0	0	-1	1	-1	1
484	0	0	-1	1	-1	1
485	0	0	-1	1	-1	1
486	0	0	1	1	1	1
487	0	0	1	1	1	1
488	0	0	1	1	1	1
489	0	0	-1	1	-1	1
490	0	0	1	1	1	1
491	0	0	-1	1	-1	1
492	0	0	-1	1	-1	1

468	2	1	0	1	1	2
469	2	1	0	1	-1	2
470	2	1	0	1	-1	2
471	2	1	0	1	-1	2
472	2	1	0	1	1	2
473	2	1	0	1	-1	2
474	2	1	0	1	-1	2
475	2	1	0	1	-1	2
476	2	1	0	1	-1	2
477	2	1	0	1	-1	2
478	2	1	0	1	-1	2
479	2	1	0	1	-1	2
480	2	1	0	1	-1	2
481	2	1	0	1	-1	2
482	2	1	0	1	-1	2
483	2	1	0	1	-1	2
484	2	1	0	1	-1	2
485	2	1	0	1	-1	2
486	2	1	0	1	1	2
487	2	1	0	1	1	2
488	2	1	0	1	1	2
489	2	1	0	1	-1	2
490	2	1	0	1	1	2
491	2	1	0	1	-1	2
492	2	1	0	1	-1	2

493	0	0	1	1	1	1
494	0	0	-1	1	-1	1
495	0	0	-1	1	-1	1
496	0	0	-1	1	-1	1
497	0	0	1	1	1	1
498	0	0	-1	1	-1	1
499	0	0	-1	1	-1	1
500	0	0	-1	1	-1	1
501	0	0	-1	1	-1	1
502	0	0	-1	1	-1	1
503	0	0	-1	1	-1	1
504	0	0	-1	1	-1	1
505	0	0	-1	1	-1	1
506	0	0	-1	1	-1	1
507	0	0	-1	1	-1	1
508	0	0	-1	1	-1	1
509	0	0	-1	1	-1	1
510	0	0	-1	1	-1	1
511	0	0	-1	1	-1	1
512	0	0	1	1	1	1
513	0	0	1	1	1	1
514	0	0	1	1	1	1
515	0	0	-1	1	-1	1
516	0	0	-1	1	-1	1
517	0	0	-1	1	-1	1

493	2	1	0	1	1	2
494	2	1	0	1	-1	2
495	2	1	0	1	-1	2
496	2	1	0	1	-1	2
497	2	1	0	1	1	2
498	2	1	0	1	-1	2
499	2	1	0	1	-1	2
500	2	1	0	1	-1	2
501	2	1	0	1	-1	2
502	2	1	0	1	-1	2
503	2	1	0	1	-1	2
504	2	1	0	1	-1	2
505	2	1	0	1	-1	2
506	2	1	0	1	-1	2
507	2	1	0	1	-1	2
508	2	1	0	1	-1	2
509	2	1	0	1	-1	2
510	2	1	0	1	-1	2
511	2	1	0	1	-1	2
512	2	1	0	1	1	2
513	2	1	0	1	1	2
514	2	1	0	1	1	2
515	2	1	0	1	-1	2
516	2	1	0	1	-1	2
517	2	1	0	1	-1	2

518	0	0	-1	1	-1	1
519	0	0	-1	1	-1	1
520	0	0	-1	1	-1	1
521	0	0	-1	1	-1	1
522	0	0	-1	1	-1	1
523	0	0	1	1	1	1
524	0	0	-1	1	-1	1
525	0	0	-1	1	-1	1
526	0	0	-1	1	-1	1
527	0	0	-1	1	-1	1
528	0	0	-1	1	-1	1
529	0	0	-1	1	-1	1
530	0	0	-1	1	-1	1
531	0	0	-1	1	-1	1
532	0	0	-1	1	-1	1
533	0	0	-1	1	-1	1
534	0	0	-1	1	-1	1
535	0	0	-1	1	-1	1
536	0	0	-1	1	-1	1
537	0	0	1	1	1	1
538	0	0	1	1	1	1
539	0	0	1	1	1	1
540	0	0	-1	1	-1	1
541	0	0	-1	1	-1	1
542	0	0	-1	1	-1	1

518	2	1	0	1	-1	2
519	2	1	0	1	-1	2
520	2	1	0	1	-1	2
521	2	1	0	1	-1	2
522	2	1	0	1	-1	2
523	2	1	0	1	1	2
524	2	1	0	1	-1	2
525	2	1	0	1	-1	2
526	2	1	0	1	-1	2
527	2	1	0	1	-1	2
528	2	1	0	1	-1	2
529	2	1	0	1	-1	2
530	2	1	0	1	-1	2
531	2	1	0	1	-1	2
532	2	1	0	1	-1	2
533	2	1	0	1	-1	2
534	2	1	0	1	-1	2
535	2	1	0	1	-1	2
536	2	1	0	1	-1	2
537	2	1	0	1	1	2
538	2	1	0	1	1	2
539	2	1	0	1	1	2
540	2	1	0	1	-1	2
541	2	1	0	1	-1	2
542	2	1	0	1	-1	2

543	0	0	-1	1	-1	1
544	0	0	-1	1	-1	1
545	0	0	-1	1	-1	1
546	0	0	-1	1	-1	1
547	0	0	1	1	1	1
548	0	0	1	1	1	1
549	0	0	-1	1	-1	1
550	0	0	-1	1	-1	1
551	0	0	-1	1	-1	1
552	0	0	-1	1	-1	1
553	0	0	-1	1	-1	1
554	0	0	-1	1	-1	1
555	0	0	-1	1	-1	1
556	0	0	-1	1	-1	1
557	0	0	-1	1	-1	1
558	0	0	-1	1	-1	1
559	0	0	-1	1	-1	1
560	0	0	-1	1	-1	1
561	0	0	-1	1	-1	1
562	0	0	-1	1	-1	1
563	0	0	1	1	1	1
564	0	0	-1	1	-1	1
565	0	0	1	1	1	1
566	0	0	1	1	1	1
567	0	0	-1	1	-1	1

543	2	1	0	1	-1	2
544	2	1	0	1	-1	2
545	2	1	0	1	-1	2
546	2	1	0	1	-1	2
547	2	1	0	1	1	2
548	2	1	0	1	1	2
549	2	1	0	1	-1	2
550	2	1	0	1	-1	2
551	2	1	0	1	-1	2
552	2	1	0	1	-1	2
553	2	1	0	1	-1	2
554	2	1	0	1	-1	2
555	2	1	0	1	-1	2
556	2	1	0	1	-1	2
557	2	1	0	1	-1	2
558	2	1	0	1	-1	2
559	2	1	0	1	-1	2
560	2	1	0	1	-1	2
561	2	1	0	1	-1	2
562	2	1	0	1	-1	2
563	2	1	0	1	1	2
564	2	1	0	1	-1	2
565	2	1	0	1	1	2
566	2	1	0	1	1	2
567	2	1	0	1	-1	2

568	0	0	-1	1	-1	1
569	0	0	-1	1	-1	1
570	0	0	-1	1	-1	1
571	0	0	1	1	1	1
572	0	0	1	1	1	1
573	0	0	-1	1	-1	1
574	0	0	-1	1	-1	1
575	0	0	-1	1	-1	1
576	0	0	-1	1	-1	1
577	0	0	-1	1	-1	1
578	0	0	-1	1	-1	1
579	0	0	-1	1	-1	1
580	0	0	-1	1	-1	1
581	0	0	-1	1	-1	1
582	0	0	-1	1	-1	1
583	0	0	-1	1	-1	1
584	0	0	-1	1	-1	1
585	0	0	-1	1	-1	1
586	0	0	-1	1	-1	1
587	0	0	-1	1	-1	1
588	0	0	-1	1	-1	1
589	0	0	-1	1	-1	1
590	0	0	-1	1	-1	1
591	0	0	1	1	1	1
592	0	0	1	1	1	1

568	2	1	0	1	-1	2
569	2	1	0	1	-1	2
570	2	1	0	1	-1	2
571	2	1	0	1	1	2
572	2	1	0	1	1	2
573	2	1	0	1	-1	2
574	2	1	0	1	-1	2
575	2	1	0	1	-1	2
576	2	1	0	1	-1	2
577	2	1	0	1	-1	2
578	2	1	0	1	-1	2
579	2	1	0	1	-1	2
580	2	1	0	1	-1	2
581	2	1	0	1	-1	2
582	2	1	0	1	-1	2
583	2	1	0	1	-1	2
584	2	1	0	1	-1	2
585	2	1	0	1	-1	2
586	2	1	0	1	-1	2
587	2	1	0	1	-1	2
588	2	1	0	1	-1	2
589	2	1	0	1	-1	2
590	2	1	0	1	-1	2
591	2	1	0	1	1	2
592	2	1	0	1	1	2

593	0	0	1	1	1	1
594	0	0	1	1	1	1
595	0	0	1	1	1	1
596	0	0	1	1	1	1
597	0	0	-1	1	-1	1
598	0	0	-1	1	-1	1
599	0	0	-1	1	-1	1
600	0	0	-1	1	-1	1
601	0	0	-1	1	-1	1
602	0	0	-1	1	-1	1
603	0	0	-1	1	-1	1
604	0	0	-1	1	-1	1
605	0	0	-1	1	-1	1
606	0	0	-1	1	-1	1
607	0	0	-1	1	-1	1
608	0	0	-1	1	-1	1
609	0	0	-1	1	-1	1
610	0	0	-1	1	-1	1
611	0	0	-1	1	-1	1
612	0	0	-1	1	-1	1
613	0	0	-1	1	-1	1
614	0	0	-1	1	-1	1
615	0	0	-1	1	-1	1
616	0	0	-1	1	-1	1
617	0	0	-1	1	-1	1

593	2	1	0	1	1	2
594	2	1	0	1	1	2
595	2	1	0	1	1	2
596	2	1	0	1	1	2
597	2	1	0	1	-1	2
598	2	1	0	1	-1	2
599	2	1	0	1	-1	2
600	2	1	0	1	-1	2
601	2	1	0	1	-1	2
602	2	1	0	1	-1	2
603	2	1	0	1	-1	2
604	2	1	0	1	-1	2
605	2	1	0	1	-1	2
606	2	1	0	1	-1	2
607	2	1	0	1	-1	2
608	2	1	0	1	-1	2
609	2	1	0	1	-1	2
610	2	1	0	1	-1	2
611	2	1	0	1	-1	2
612	2	1	0	1	-1	2
613	2	1	0	1	-1	2
614	2	1	0	1	-1	2
615	2	1	0	1	-1	2
616	2	1	0	1	-1	2
617	2	1	0	1	-1	2

618	0	0	-1	1	-1	1
619	0	0	-1	1	-1	1
620	0	0	-1	1	-1	1
621	0	0	-1	1	-1	1
622	0	0	-1	1	-1	1
623	0	0	-1	1	-1	1
624	0	0	-1	1	-1	1
625	0	0	-1	1	-1	1

618	2	1	0	1	-1	2
619	2	1	0	1	-1	2
620	2	1	0	1	-1	2
621	2	1	0	1	-1	2
622	2	1	0	1	-1	2
623	2	1	0	1	-1	2
624	2	1	0	1	-1	2
625	2	1	0	1	-1	2

DATA UJI (KENDARAAN RODA DUA)

NO	DATA UJI	BOBOT AKHIR	HASIL KALI	TOTAL	HASIL
1	-1	0	0	122	1
2	-1	0	0		
3	-1	0	0		
4	-1	0	0		
5	-1	0	0		
6	-1	0	0		
7	-1	0	0		
8	-1	0	0		
9	-1	0	0		
10	-1	0	0		
11	-1	0	0		
12	-1	0	0		
13	-1	0	0		
14	-1	0	0		
15	-1	0	0		
16	-1	0	0		
17	-1	0	0		
18	-1	0	0		
19	-1	0	0		
20	-1	0	0		
21	-1	0	0		
22	-1	0	0		
23	-1	0	0		
24	-1	0	0		
25	-1	0	0		
26	-1	0	0		
27	-1	0	0		
28	-1	0	0		
29	-1	0	0		
30	-1	0	0		
31	-1	0	0		
32	-1	0	0		
33	-1	0	0		
34	-1	0	0		
35	1	0	0		
36	-1	2	-2		
37	-1	0	0		
38	-1	0	0		
39	1	2	2		
40	-1	0	0		
41	-1	0	0		
42	1	0	0		
43	1	0	0		
44	-1	0	0		
45	-1	0	0		
46	-1	0	0		

47	-1	0	0		
48	-1	0	0		
49	-1	0	0		
50	-1	0	0		
51	-1	0	0		
52	-1	0	0		
53	-1	0	0		
54	-1	0	0		
55	-1	0	0		
56	-1	0	0		
57	-1	2	-2		
58	-1	0	0		
59	-1	0	0		
60	1	2	2		
61	1	2	2		
62	1	0	0		
63	1	2	2		
64	-1	0	0		
65	1	-2	-2		
66	1	0	0		
67	-1	0	0		
68	1	-2	-2		
69	1	-2	-2		
70	1	0	0		
71	1	0	0		

72	-1	0	0		
73	-1	0	0		
74	-1	0	0		
75	-1	0	0		
76	-1	0	0		
77	-1	0	0		
78	-1	0	0		
79	-1	0	0		
80	-1	0	0		
81	-1	0	0		
82	-1	2	-2		
83	-1	2	-2		
84	-1	0	0		
85	1	2	2		
86	1	0	0		
87	1	2	2		
88	-1	0	0		
89	1	0	0		
90	1	2	2		
91	1	0	0		
92	-1	-2	2		
93	-1	-2	2		
94	-1	2	-2		
95	-1	-2	2		
96	1	-2	-2		

97	1	0	0		
98	-1	0	0		
99	-1	0	0		
100	-1	0	0		
101	-1	0	0		
102	-1	0	0		
103	-1	0	0		
104	-1	0	0		
105	-1	0	0		
106	-1	0	0		
107	-1	2	-2		
108	-1	2	-2		
109	-1	2	-2		
110	1	0	0		
111	1	2	2		
112	-1	0	0		
113	-1	-2	2		
114	1	2	2		
115	-1	2	-2		
116	-1	2	-2		
117	1	0	0		
118	-1	-2	2		
119	1	2	2		
120	-1	-2	2		
121	-1	-2	2		

122	1	0	0		
123	-1	0	0		
124	-1	0	0		
125	-1	0	0		
126	-1	0	0		
127	-1	0	0		
128	-1	0	0		
129	-1	0	0		
130	-1	0	0		
131	-1	0	0		
132	-1	2	-2		
133	-1	2	-2		
134	-1	2	-2		
135	1	0	0		
136	1	0	0		
137	1	2	2		
138	-1	2	-2		
139	1	2	2		
140	1	0	0		
141	1	2	2		
142	1	0	0		
143	-1	-2	2		
144	1	2	2		
145	-1	-2	2		
146	1	-2	-2		

147	1	0	0		
148	-1	0	0		
149	-1	0	0		
150	-1	0	0		
151	-1	0	0		
152	-1	0	0		
153	-1	0	0		
154	-1	0	0		
155	-1	0	0		
156	-1	0	0		
157	-1	-2	2		
158	-1	2	-2		
159	-1	2	-2		
160	1	0	0		
161	1	0	0		
162	-1	2	-2		
163	-1	2	-2		
164	1	2	2		
165	1	2	2		
166	1	2	2		
167	-1	0	0		
168	1	0	0		
169	1	0	0		
170	-1	-2	2		
171	1	0	0		

172	1	0	0		
173	-1	0	0		
174	-1	0	0		
175	-1	0	0		
176	-1	0	0		
177	-1	0	0		
178	-1	0	0		
179	-1	0	0		
180	-1	-2	2		
181	-1	0	0		
182	-1	-2	2		
183	-1	2	-2		
184	-1	0	0		
185	-1	0	0		
186	1	2	2		
187	1	-2	-2		
188	1	2	2		
189	1	2	2		
190	-1	2	-2		
191	-1	-2	2		
192	1	0	0		
193	1	0	0		
194	1	-2	-2		
195	1	-2	-2		
196	-1	0	0		

197	-1	0	0		
198	-1	0	0		
199	-1	0	0		
200	-1	0	0		
201	-1	0	0		
202	-1	2	-2		
203	-1	0	0		
204	-1	-2	2		
205	-1	0	0		
206	-1	0	0		
207	-1	-2	2		
208	-1	2	-2		
209	-1	0	0		
210	-1	2	-2		
211	1	2	2		
212	1	-2	-2		
213	1	2	2		
214	1	2	2		
215	1	0	0		
216	1	0	0		
217	1	0	0		
218	1	0	0		
219	1	-2	-2		
220	1	-2	-2		
221	-1	-2	2		

222	-1	0	0		
223	-1	0	0		
224	-1	0	0		
225	-1	0	0		
226	-1	0	0		
227	-1	2	-2		
228	-1	0	0		
229	-1	0	0		
230	-1	0	0		
231	-1	0	0		
232	-1	-2	2		
233	-1	0	0		
234	-1	2	-2		
235	-1	2	-2		
236	1	2	2		
237	1	2	2		
238	1	0	0		
239	1	2	2		
240	1	0	0		
241	1	-2	-2		
242	1	2	2		
243	1	2	2		
244	1	0	0		
245	1	-2	-2		
246	-1	-2	2		

247	-1	0	0		
248	-1	0	0		
249	-1	0	0		
250	-1	0	0		
251	-1	0	0		
252	-1	0	0		
253	-1	0	0		
254	-1	0	0		
255	-1	-2	2		
256	-1	-2	2		
257	-1	-2	2		
258	-1	0	0		
259	-1	2	-2		
260	-1	2	-2		
261	1	2	2		
262	1	2	2		
263	1	0	0		
264	1	0	0		
265	1	0	0		
266	1	-2	-2		
267	1	0	0		
268	1	2	2		
269	1	-2	-2		
270	1	-2	-2		
271	-1	-2	2		

272	-1	0	0		
273	-1	0	0		
274	-1	0	0		
275	-1	0	0		
276	-1	0	0		
277	-1	-2	2		
278	-1	-2	2		
279	-1	-2	2		
280	-1	2	-2		
281	-1	0	0		
282	-1	-2	2		
283	-1	0	0		
284	-1	0	0		
285	-1	2	-2		
286	-1	0	0		
287	1	0	0		
288	-1	0	0		
289	1	0	0		
290	-1	0	0		
291	1	0	0		
292	1	2	2		
293	1	2	2		
294	1	0	0		
295	1	0	0		
296	-1	-2	2		

297	-1	-2	2		
298	-1	-2	2		
299	-1	0	0		
300	-1	0	0		
301	-1	0	0		
302	-1	-2	2		
303	1	-2	-2		
304	-1	0	0		
305	-1	0	0		
306	-1	0	0		
307	-1	-2	2		
308	-1	0	0		
309	-1	0	0		
310	-1	0	0		
311	-1	0	0		
312	-1	-2	2		
313	-1	2	-2		
314	-1	2	-2		
315	-1	2	-2		
316	-1	-2	2		
317	1	-2	-2		
318	1	0	0		
319	1	-2	-2		
320	-1	-2	2		
321	-1	-2	2		

322	-1	-2	2		
323	-1	-2	2		
324	-1	-2	2		
325	-1	0	0		
326	-1	0	0		
327	-1	-2	2		
328	-1	-2	2		
329	1	-2	-2		
330	-1	0	0		
331	-1	0	0		
332	1	-2	-2		
333	-1	0	0		
334	-1	0	0		
335	-1	0	0		
336	-1	0	0		
337	-1	0	0		
338	-1	2	-2		
339	-1	0	0		
340	-1	0	0		
341	-1	0	0		
342	-1	0	0		
343	-1	-2	2		
344	-1	0	0		
345	-1	-2	2		
346	-1	0	0		

347	-1	0	0		
348	-1	-2	2		
349	-1	-2	2		
350	-1	0	0		
351	-1	0	0		
352	-1	-2	2		
353	-1	-2	2		
354	-1	0	0		
355	-1	-2	2		
356	1	2	2		
357	1	2	2		
358	-1	-2	2		
359	-1	0	0		
360	-1	0	0		
361	-1	-2	2		
362	-1	0	0		
363	-1	0	0		
364	-1	0	0		
365	1	0	0		
366	1	0	0		
367	1	0	0		
368	-1	0	0		
369	-1	-2	2		
370	-1	-2	2		
371	-1	-2	2		

372	-1	-2	2		
373	-1	0	0		
374	-1	0	0		
375	-1	0	0		
376	-1	0	0		
377	-1	-2	2		
378	-1	-2	2		
379	-1	0	0		
380	-1	-2	2		
381	1	0	0		
382	1	0	0		
383	1	-2	-2		
384	1	0	0		
385	1	0	0		
386	-1	-2	2		
387	-1	0	0		
388	1	0	0		
389	-1	2	-2		
390	1	2	2		
391	1	0	0		
392	1	0	0		
393	1	0	0		
394	1	-2	-2		
395	-1	0	0		
396	-1	0	0		

397	-1	0	0		
398	-1	0	0		
399	-1	0	0		
400	-1	0	0		
401	-1	0	0		
402	-1	0	0		
403	-1	0	0		
404	-1	2	-2		
405	-1	0	0		
406	1	0	0		
407	1	0	0		
408	1	0	0		
409	-1	0	0		
410	1	-2	-2		
411	1	0	0		
412	1	0	0		
413	-1	0	0		
414	1	2	2		
415	1	2	2		
416	1	2	2		
417	1	2	2		
418	1	0	0		
419	1	2	2		
420	-1	0	0		
421	-1	0	0		

422	-1	0	0		
423	-1	0	0		
424	-1	0	0		
425	-1	0	0		
426	-1	0	0		
427	-1	-2	2		
428	-1	0	0		
429	-1	-2	2		
430	-1	0	0		
431	-1	0	0		
432	-1	-2	2		
433	-1	0	0		
434	-1	0	0		
435	1	-2	-2		
436	1	0	0		
437	-1	0	0		
438	1	2	2		
439	1	2	2		
440	1	2	2		
441	1	2	2		
442	1	0	0		
443	1	0	0		
444	1	0	0		
445	1	2	2		
446	-1	2	-2		

447	-1	0	0		
448	-1	0	0		
449	-1	0	0		
450	-1	0	0		
451	-1	0	0		
452	-1	0	0		
453	-1	0	0		
454	-1	-2	2		
455	-1	-2	2		
456	-1	-2	2		
457	-1	0	0		
458	-1	0	0		
459	1	-2	-2		
460	1	-2	-2		
461	1	0	0		
462	1	0	0		
463	1	2	2		
464	1	2	2		
465	1	0	0		
466	1	-2	-2		
467	-1	2	-2		
468	-1	2	-2		
469	-1	0	0		
470	-1	0	0		
471	1	0	0		

472	1	2	2		
473	-1	0	0		
474	-1	0	0		
475	-1	0	0		
476	-1	0	0		
477	-1	-2	2		
478	-1	0	0		
479	-1	0	0		
480	-1	0	0		
481	-1	-2	2		
482	-1	0	0		
483	-1	0	0		
484	-1	-2	2		
485	-1	-2	2		
486	1	0	0		
487	1	2	2		
488	1	2	2		
489	1	0	0		
490	1	0	0		
491	1	-2	-2		
492	1	0	0		
493	1	2	2		
494	1	0	0		
495	-1	0	0		
496	-1	0	0		

497	1	2	2		
498	1	0	0		
499	-1	0	0		
500	-1	0	0		
501	-1	0	0		
502	-1	0	0		
503	-1	0	0		
504	-1	-2	2		
505	-1	-2	2		
506	-1	-2	2		
507	-1	0	0		
508	-1	0	0		
509	-1	-2	2		
510	-1	-2	2		
511	-1	0	0		
512	-1	2	-2		
513	1	2	2		
514	1	0	0		
515	1	-2	-2		
516	-1	-2	2		
517	-1	-2	2		
518	1	-2	-2		
519	1	0	0		
520	-1	0	0		
521	-1	0	0		

522	-1	0	0		
523	1	2	2		
524	-1	0	0		
525	-1	0	0		
526	-1	0	0		
527	-1	0	0		
528	-1	0	0		
529	-1	0	0		
530	-1	-2	2		
531	-1	0	0		
532	-1	0	0		
533	-1	-2	2		
534	-1	-2	2		
535	-1	0	0		
536	-1	-2	2		
537	-1	0	0		
538	1	0	0		
539	1	0	0		
540	1	-2	-2		
541	-1	0	0		
542	-1	0	0		
543	-1	0	0		
544	-1	0	0		
545	-1	0	0		
546	-1	0	0		

547	-1	2	-2		
548	1	2	2		
549	-1	0	0		
550	-1	0	0		
551	-1	0	0		
552	-1	0	0		
553	-1	0	0		
554	-1	0	0		
555	-1	0	0		
556	-1	0	0		
557	-1	0	0		
558	-1	0	0		
559	-1	0	0		
560	-1	0	0		
561	-1	0	0		
562	-1	0	0		
563	-1	2	-2		
564	1	0	0		
565	1	2	2		
566	-1	2	-2		
567	-1	0	0		
568	-1	0	0		
569	-1	0	0		
570	-1	0	0		
571	-1	2	-2		

572	1	2	2		
573	1	0	0		
574	-1	0	0		
575	-1	0	0		
576	-1	0	0		
577	-1	0	0		
578	-1	0	0		
579	-1	0	0		
580	-1	0	0		
581	-1	0	0		
582	-1	0	0		
583	-1	0	0		
584	-1	0	0		
585	-1	0	0		
586	-1	0	0		
587	-1	0	0		
588	-1	0	0		
589	-1	0	0		
590	-1	0	0		
591	1	2	2		
592	1	2	2		
593	-1	2	-2		
594	-1	2	-2		
595	-1	2	-2		
596	1	2	2		

597	1	0	0		
598	-1	0	0		
599	-1	0	0		
600	-1	0	0		
601	-1	0	0		
602	-1	0	0		
603	-1	0	0		
604	-1	0	0		
605	-1	0	0		
606	-1	0	0		
607	-1	0	0		
608	-1	0	0		
609	-1	0	0		
610	-1	0	0		
611	-1	0	0		
612	-1	0	0		
613	-1	0	0		
614	-1	0	0		
615	-1	0	0		
616	-1	0	0		
617	1	0	0		
618	1	0	0		
619	1	0	0		
620	1	0	0		
621	-1	0	0		
622	-1	0	0		
623	-1	0	0		
624	-1	0	0		
625	-1	0	0		

LAMPIRAN B DATA UJI KENDARAAN RODA EMPAT

NO	Kendaraan Roda Empat	Manual	Sistem
1		Positif	Positif
2		Positif	Positif
3		Positif	Positif
4		Positif	Negatif
5		Positif	Negatif
6		Positif	Positif
7		Positif	Positif
8		Positif	Positif
9		Positif	Positif
10		Positif	Positif
11		Positif	Positif
12		Positif	Positif
13		Positif	Positif
14		Positif	Positif
15		Positif	Positif
16		Positif	Negatif
17		Positif	Negatif
18		Positif	Positif
19		Positif	Positif
20		Positif	Negatif
21		Positif	Positif
22		Positif	Positif
23		Positif	Positif
24		Positif	Negatif
25		Positif	Positif

DATA UJI KENDARAAN RODA DUA

No	Kendaraan Roda Dua	Manual	Sistem
1.		Positif	Positif
2.		Positif	Positif
3.		Positif	Positif
4.		Positif	Positif
5.		Positif	Positif
6		Positif	Negatif
7.		Positif	Positif
8.		Positif	Positif
9.		Positif	Positif
10.		Positif	Positif
11.		Positif	Positif
12.		Positif	Positif
13.		Positif	Negatif
14.		Positif	Positif
15.		Positif	Positif
16.		Positif	Negatif
17.		Positif	Positif
18.		Positif	Positif
19.		Positif	Positif
20.		Positif	Negatif
21.		Positif	Negatif
22.		Positif	Positif
23.		Positif	Positif
24.		Positif	Positif
25.		Positif	Positif
26.		Positif	Positif
27.		Positif	Positif
28.		Positif	Positif

29.		Positif	Positif
30.		Positif	Positif