



TESIS- TF185471

**PERANCANGAN ALGORITMA KONTROL PADA
SISTEM MONITORING DAN KONTROL
JEMBATAN BERBASIS *INTERNET OF THINGS*
(IoT) SECARA *REAL TIME***

FARIDA IASHA
NRP. 02311850010006

DOSEN PEMBIMBING
Dr.Ir. Purwadi Agus Darwito,M.Sc.

Departemen Teknik Fisika
Fakultas Teknologi Industri dan Rekayasa Sistem
Institut Teknologi Sepuluh Nopember
2020

LEMBAR PENGESAHAN TESIS

Telah disusun untuk memenuhi salah satu syarat memperoleh gelar

Magister Teknik (MT)

di

Institut Teknologi Sepuluh Nopember

Oleh:

FARIDA IASHA

NRP: 02311850010006

Tanggal Ujian : 6 Januari 2020

Periode Wisuda : Maret 2020

Disetujui oleh :

Pembimbing

1. Dr. Ir. Purwadi Agus Darwito, M.Sc.

NIP: 19620822 198803 1 001

Penguji

2. Dr. Suyanto, S.T., M.T.

NIP: 19711113 199512 1 002

3. Dr. Ir. Syamsul Arifin, M.T.

NIP: 19630907 198903 1 004



Kepala Departemen Teknik Fisika
Fakultas Teknologi Industri dan Rekayasa Sistem



Dr. Suyanto, S.T., M.T.

NIP: 19711113 199512 1 002

(Halaman ini sengaja dikosongkan)

**PERANCANGAN ALGORITMA KONTROL PADA SISTEM
MONITORING DAN KONTROL JEMBATAN BERBASIS *INTERNET OF
THINGS (IoT)* SECARA *REAL TIME***

Nama Mahasiswa : Farida Iasha
NRP : 02311850010006
Pembimbing : Dr. Ir. Purwadi Agus Darwito, M.Sc.

ABSTRAK

Pembangunan infrastruktur pada transportasi darat tidak lepas dari pembangunan jembatan yang merupakan sarana penghubung antar wilayah guna menunjang kelancaran transportasi di darat. Dalam desain pembangunannya, usia jembatan umumnya di rencanakan 100 tahun untuk jembatan besar dan 50 tahun sebagai batas minimumnya. Seiring berjalannya waktu jembatan akan mengalami deformasi yang disebabkan oleh perubahan pergerakan akibat dari kondisi alam, beban jembatan, struktur konstruksi jembatan, getaran yang berlebih dan lain sebagainya. Dalam rangka mengurangi pelanggaran kelebihan muatan seperti yang telah ditetapkan diperaturan pemerintah nomor 43 tahun 1993 tentang prasarana dan lalu lintas jalan pada pasal 35 dan pasal 38 menyebutkan bahwa “Alat pengawasan dan pengaman jalan berfungsi untuk melakukan pengawasan terhadap berat kendaraan beserta muatannya, berupa alat penimbang yang dipasang secara tetap atau alat timbang yang dapat dipindah-pindahkan”. Sebagai contoh kasus pada jembatan cincin di Widang, Kabupaten Tuban yang runtuh karena beban kendaraan melebihi kapasitas kemampuan jembatan. Sehingga kasus runtuhnya jembatan di Widang mengilhami ide untuk mendata, mengawasi dan merawat setiap jembatan yang ada di Indonesia. Apabila suatu jembatan mengalami keruntuhan maka dapat mengganggu stabilitas kegiatan perekonomian di Indonesia. Algoritma yang dirancang pada sistem monitoring dan kontrol jembatan ini adalah algoritma untuk mengatur kendaraan agar kendaraan yang memasuki jembatan tidak melebihi kapasitas beban maksimal jembatan. Dalam tesis ini diasumsikan kapasitas maksimal jembatan adalah 100.000 kg. Kendaraan yang masuk jembatan telah di index oleh beberapa variabel dalam kurun waktu 24 jam. Beberapa variabel tersebut adalah waktu, identitas kendaraan berupa plat nomer, jenis kendaraan, berat

kendaraan dan index. Dalam menguji kebenaran algoritma maka digunakan perangkat lunak *visual basic*. Data kendaraan riil di persiapkan dalam data excel yang dapat mewakili data sebenarnya. Algoritma kontrol pada jembatan yang diterapkan adalah ketika kendaraan memasuki jembatan maka sensor loadcell akan mendeteksi berat kendaraan lalu sensor akselerometer mendeteksi posisi kendaraan. Data kendaraan yang melewati akan tersimpan dalam buffer dan akan dijumlahkan. Apabila kendaraan melewati sensor akselerometer ke 8 maka data kendaraan tersebut akan terhapus. Sehingga data yang tersimpan hanyalah data kendaraan yang masih ada diatas jembatan. Eksekusi penutupan portal berdasarkan hasil perhitungan dari beban kendaraan yang terukur. Apabila beban kendaraan melebihi kapasitas beban maksimal jembatan yang ditetapkan maka portal akan menutup jembatan dan sebaliknya. Informasi mengenai kondisi jembatan ditampilkan melalui website berbasis *Internet of Things (IoT)*. Penelitian perancangan algoritma control untuk sistem monitoring dan kontrol jembatan berbasis *internet of things (IoT)* secara *real time* ini bermanfaat baik dalam segi ilmu maupun aplikasi.

Kata kunci : Jembatan, Algoritma, *Internet Of Things (IoT)*, *Sistem kontrol dan monitoring* jembatan.

**DESIGN OF ALGORITHM CONTROL
FOR MONITORING SYSTEM AND CONTROL BRIDGE
BASED INTERNET OF THINGS REAL TIME**

Nama Mahasiswa : Farida Iasha
NRP : 02311850010006
Pembimbing : Dr. Ir. Purwadi Agus Darwito, M.Sc.

ABSTRACT

Infrastructure development in land transportation cannot be separated from the construction of bridges which are a means of connecting between regions to support smooth transportation on land. In its construction design, the bridge age is generally planned for 100 years for large bridges and 50 years as the minimum limit. Over time the bridge will experience deformation caused by changes in movement due to natural conditions, bridge loads, bridge construction structures, excessive vibration and so forth. In order to reduce overload violations as stipulated in government regulation number 43 of 1993 concerning road infrastructure and traffic in Article 35 and Article 38 states that "Road surveillance and safety devices function to supervise the weight of vehicles and their contents, in the form of weighing devices which fixed or mobile weighing devices ". For example the case of the ring bridge in Widang, Tuban District which collapsed because the vehicle load exceeded the bridge's capacity. So that the bridge collapse case in Widang inspired the idea to collect, monitor and maintain every bridge in Indonesia. If a bridge collapses, it can disrupt the stability of economic activity in Indonesia. The algorithm designed on the bridge monitoring and control system is an algorithm to regulate the vehicle so that the vehicle entering the bridge does not exceed the bridge's maximum load capacity. In this thesis the maximum bridge capacity is assumed to be 100,000 kg. Vehicles entering the bridge have been indexed by several variables within 24 hours. Some of these variables are time, vehicle identity in the form of license plates, type of vehicle, vehicle weight and index. In testing the truth of the algorithm, visual basic software is used. Real vehicle data is prepared in excel data that can represent the actual data. The

control algorithm applied on the bridge is when the vehicle enters the bridge, the loadcell sensor will detect the weight of the vehicle and the accelerometer sensor detects the vehicle's position. Vehicle data that passes will be stored in a buffer and will be added together. If the vehicle passes the 8th accelerometer sensor, the vehicle data will be erased. So that the data stored is only vehicle data that is still on the bridge. Portal closure execution is based on the calculation of the measured vehicle load. If the vehicle load exceeds the maximum bridge load capacity specified then the portal will close the bridge and vice versa. Information about the condition of the bridge is displayed through the internet-based website of Things (IoT). This research on the design of control algorithms for bridge monitoring and control systems based on internet of things (IoT) in real time is useful both in terms of science and application.

Keywords: *Bridges, Algorithms, Internet of Things (IoT), bridge control and monitoring systems.*

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah S.W.T, karena rahmat-Nya hingga penulis diberikan kesehatan, kemudahan, kelancaran dalam menyusun laporan tesis yang berjudul “ Perancangan Algoritma Kontrol pada Sistem Monitoring dan Kontrol Jembatan Berbasis *Internet of Things (IoT)* secara Real Time”. Pada kesempatan ini, penulis hendak mengucapkan terima kasih kepada:

1. Dr. Suyanto,S.T.,M.T. selaku Kepala Departemen Teknik Fisika ITS.
2. Dr. Ir. Purwadi Agus Darwito, M.Sc. selaku dosen pembimbing yang telah memberikan banyak ilmu dan membantu dalam proses pengerjaan tesis.
3. Segenap Bapak/ibu dosen, khususnya pengampu bidang minat rekayasa instrumentasi di Departemen Teknik Fisika ITS yang telah memberikan banyak ilmu penulis selama diperkuliahan.
4. Orang tua dan keluarga penulis yang senantiasa memberikan doa, bantuan dan motivasi.
5. Pascasarjana Teknik Fisika ITS angkatan 2018 yang saling memberikan semangat, doa serta selalu menghibur.
6. Sahabat-sahabat penulis yang telah mendoakan untuk kelancaran tesis dan pihak lain yang telah membantu penulis menyelesaikan tesis ini.

Penulis mengucapkan mohon maaf atas ketidaksempurnaan dalam penulisan laporan tesis ini. Semoga laporan tesis ini bermanfaat bagi pembaca.

Surabaya, 10 Januari 2020

(Halaman ini sengaja dikosongkan)

DAFTAR ISI

LEMBAR PENGESAHAN TESIS.....	Error! Bookmark not defined.
ABSTRAK	v
ABSTRACT.....	vii
KATA PENGANTAR	ix
DAFTAR ISI.....	xi
DAFTAR GAMBAR	xiii
DAFTAR TABEL.....	xv
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	6
1.3. Tujuan Penelitian	6
1.4 Batasan Masalah	6
1.5 Manfaat Penelitian	7
BAB 2 TINJAUAN PUSTAKA	9
2.1 Jembatan	9
2.2 Diagram Fishbone Penelitian	15
2.3 Sistem Monitoring dan Kontrol	19
2.4 <i>Internet Of things(IoT)</i>	22
2.5 <i>Wireless Sensor Network (WSN)</i>	23
2.6 Visual Basic Net(VBnet)	25
2.7 Algoritma.....	25
BAB 3 METODOLOGI PENELITIAN.....	27
3.1 Prosedur Penelitian	27
3.2 Perancangan sistem.....	29
3.2.1 Perancangan Desain Jembatan.....	29
3.2.2 Perancangan Algoritma dalam mengatur beban jembatan.....	31
3.2.3 Perancangan konsep sistem monitoring dan kontrol jembatan secara hardware.....	40
3.2.4 Visualisasi alur konsep algoritma pada Simulasi Sistem Monitoring dan kontrol Jembatan.....	45

BAB 4 ANALISA DATA DAN PEMBAHASAN	51
4.1 Pengujian Algoritma.....	51
4.2 Validasi Algoritma	79
BAB 5 KESIMPULAN DAN SARAN	84
DAFTAR PUSTAKA.....	87

DAFTAR GAMBAR

Gambar 2.1 Jembatan Suramadu (Direktorat Jembatan Bina Marga, 2016).....	10
Gambar 2. 2 Jembatan lantai bawah (Rahadi, 2015).....	11
Gambar 2. 3 Jembatan lantai Atas (Rahadi, 2015).....	12
Gambar 2. 4 Jembatan rangka tipe Warren with verticals (Rahadi, 2015).....	12
Gambar 2. 5 Jembatan rangka tipe <i>howe</i> (Rahadi, 2015).....	12
Gambar 2. 6 Struktur Jembatan(Zhao et al., 2011)	14
Gambar 2. 7 The sensors arrangement map of J2 section and J4 section(Zhao et al., 2011)	14
Gambar 2. 8 The sensors arrangement map of J2 section and J4 section(Zhao et al., 2011)	14
Gambar 2. 9 Fishbone penelitian sistem monitoring dan kontrol jembatan.....	16
Gambar 2. 10 Internet of Things(Meutia, 2015)	22
Gambar 2. 11 Blok Sistem <i>Internet Of Things</i> (Meutia, 2015)	23
Gambar 2. 12 Arsitektur WSN(Moulat et al., 2018)	24
Gambar 2. 13 Komponen-komponen yang terhubung dengan node (Li & Kara, 2017).....	24
Gambar 3. 1 Diagram Alir Penelitian.....	28
Gambar 3. 2 Desain jembatan	29
Gambar 3. 3 Desain peletakan sensor Akselerometer	30
Gambar 3. 4 Flowchart Algoritma Kontrol Jembatan	Error! Bookmark not defined.
Gambar 3. 5 Arsitektur Hardware (Purbo, 2017).....	40
Gambar 3. 6 Koneksi antara Rapsberry , internet dan server	43
Gambar 3. 7 Koneksi antar perangkat	44
Gambar 3. 8 Protokol <i>Internet of Things (IoT)</i>	45
Gambar 3. 9 Tampilan Xampp Control Panel	47
Gambar 3. 10 Tampilan data MySQL	47
Gambar 3. 11 Tampilan Website.....	48
Gambar 3. 12 Tampilan Website pada monitoring kendaraan lajur kiri	48
Gambar 3. 13 Tampilan Website pada monitoring kendaraan lajur kanan	49
Gambar 3. 14 Desain HMI pada simulasi sistem jembatan.....	49
Gambar 3. 15 Hubungan antara server, database dan client.....	50
Gambar 4. 1 Jembatan dalam kondisi lengang.....	53
Gambar 4. 2 Jembatan dalam keadaan macet.....	55
Gambar 4. 3 Jembatan dalam kondisi normal	57
Gambar 4. 4 Jembatan dalam kondisi lengang.....	60
Gambar 4. 5 Tampilan informasi kendaraan pada lajur kanan.....	61
Gambar 4. 6 Tampilan informasi kendaraan pada lajur kanan.....	61

Gambar 4. 7	Jembatan dalam kondisi lengang	62
Gambar 4. 8	Tampilan informasi kendaraan pada lajur kiri	62
Gambar 4. 9	Tampilan informasi kendaraan pada lajur kanan	63
Gambar 4. 10	Pengujian algoritma terhadap kondisi jembatan lengang	63
Gambar 4. 11	Tampilan informasi kendaraan pada lajur kiri	64
Gambar 4. 12	Tampilan informasi kendaraan pada lajur kanan	64
Gambar 4. 13	Pengujian algoritma terhadap kondisi jembatan lengang	65
Gambar 4. 14	Tampilan informasi kendaraan pada lajur kiri	65
Gambar 4. 15	Tampilan informasi kendaraan pada lajur kanan	66
Gambar 4. 16	Pengujian algoritma terhadap kondisi jembatan macet.....	67
Gambar 4. 17	Tampilan informasi kendaraan pada lajur kanan	67
Gambar 4. 18	Tampilan informasi kendaraan pada lajur kiri	68
Gambar 4. 19	Pengujian algoritma terhadap kondisi jembatan macet.....	68
Gambar 4. 20	Tampilan informasi kendaraan pada lajur kiri	69
Gambar 4. 21	Tampilan informasi kendaraan pada lajur kanan	70
Gambar 4. 22	Pengujian algoritma terhadap kondisi jembatan macet.....	70
Gambar 4. 23	Tampilan informasi kendaraan pada lajur kiri	71
Gambar 4. 24	Tampilan informasi kendaraan pada lajur kanan	71
Gambar 4. 25	Pengujian algoritma terhadap kondisi jembatan normal.....	72
Gambar 4. 26	Tampilan informasi kendaraan pada lajur kiri	73
Gambar 4. 27	Tampilan informasi kendaraan pada lajur kiri	73
Gambar 4. 28	Pengujian algoritma terhadap kondisi jembatan normal.....	74
Gambar 4. 29	Tampilan informasi kendaraan pada lajur kiri	74
Gambar 4. 30	Tampilan informasi kendaraan pada lajur kanan	75
Gambar 4. 31	Pengujian algoritma dengan data truk.....	75
Gambar 4. 32	Tampilan informasi kendaraan pada lajur kiri	76
Gambar 4. 33	Tampilan informasi kendaraan pada lajur kanan	76
Gambar 4. 34	Pengujian algoritma dengan data kendaraan pribadi	77
Gambar 4. 35	Tampilan informasi kendaraan pada lajur kiri	77
Gambar 4. 36	Tampilan informasi kendaraan pada lajur kanan	77
Gambar 4. 37	Pengujian algoritma dengan data bus.....	78
Gambar 4. 38	Tampilan informasi kendaraan pada lajur kanan	78
Gambar 4. 39	Tampilan informasi kendaraan pada lajur kiri	79
Gambar 4. 40	Algoritma first come first served.	80

DAFTAR TABEL

Tabel 3. 1 Ilustrasi proses algoritma kontrol jembatan berprinsip <i>First in first out</i> (FIFO).....	33
Tabel 3. 2 Ilustrasi proses algoritma kontrol pada kondisi lengang	37
Tabel 4. 1 Daftar Proses Antrian FCFS.....	80
Tabel 4. 2 Algoritma FCFS dan Algoritma penelitian pada saat pembacaan berat kendaraan oleh loadcell	81
Tabel 4. 3 Proses pembacaan posisi kendaraan dengan sensor akselerometer saat jembatan dalam kondisi jembatan macet.....	81
Tabel 4. 4 Proses pembacaan posisi kendaraan dengan sensor akselerometer saat jembatan dalam kondisi jembatan macet.....	82
Tabel 4. 5 Proses dalam komputer beserta waktu pelayanan	83

(Halaman ini sengaja dikosongkan)

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Transportasi memegang peran penting dalam pertumbuhan ekonomi suatu negara. Infrastruktur transportasi darat yang utama adalah jalan dan jembatan. Pembangunan infrastruktur jalan dan jembatan mempunyai peranan penting dalam meningkatkan kesejahteraan masyarakat dan pertumbuhan ekonomi suatu daerah (Effendi & Hendarto, 2014). Saat ini pemerintah berupaya untuk meningkatkan infrastruktur jalur darat baik dalam bentuk tol maupun jembatan. Pembangunan jembatan di Indonesia mengalami kenaikan di setiap tahunnya mulai 2015 sampai 2018 seperti yang direncanakan oleh presiden (Effendi & Hendarto, 2014). Salah satu pembangunan infrastruktur adalah pembangunan jembatan yang merupakan sarana penghubung antar wilayah yang terpisah oleh sungai atau lembah. Dalam desain pembangunannya, usia jembatan umumnya di rencanakan 100 tahun untuk jembatan besar dan 50 tahun sebagai batas minimumnya.

Seiring berjalannya waktu jembatan akan mengalami deformasi yang disebabkan oleh perubahan pergerakan akibat dari kondisi alam, beban jembatan, struktur konstruksi jembatan, getaran yang berlebih dan lain sebagainya (Kuang, 1996). Salah satu penyebab jembatan runtuh adalah beban yang melebihi kapasitas (Yadika, 2018). Dalam rangka mengurangi pelanggaran kelebihan muatan seperti yang telah ditetapkan diperaturan pemerintah nomor 43 tahun 1993 tentang prasarana dan lalu lintas jalan pada pasal 35 dan pasal 38 menyebutkan bahwa “Alat pengawasan dan pengaman jalan berfungsi untuk melakukan pengawasan terhadap berat kendaraan beserta muatannya, berupa alat penimbang yang dipasang secara tetap atau alat timbang yang dapat dipindah-pindahkan”. Sebagai contoh kasus pada jembatan cincin di Widang, Kabupaten Tuban yang runtuh karena beban kendaraan melebihi kapasitas kemampuan jembatan. Sehingga kasus runtuhnya jembatan di Widang mengilhami ide untuk mendata, mengawasi dan merawat setiap jembatan yang ada di Indonesia. Apabila suatu jembatan mengalami keruntuhan maka dapat mengganggu stabilitas kegiatan perekonomian di Indonesia.

Salah satu permasalahan di Indonesia yang berhubungan dengan perubahan deformasi yaitu jembatan suramadu (Maris, 2018). Jembatan Suramadu telah dibangun dengan investasi yang cukup besar, maka dari itu diharapkan jembatan suramadu memiliki umur yang panjang yaitu 100 tahun. Namun pada kenyataannya, setiap hari jembatan Suramadu dilalui 35.000 sepeda motor serta 23.000 mobil (Maris, 2018). Hal ini menyebabkan struktur jembatan berpotensi mengalami deformasi yang dapat menimbulkan penurunan kemampuan jembatan dalam menopang beban kendaraan. Penelitian tersebut membahas tentang penurunan kemampuan degradasi dan kerusakan *deterioration* pada jembatan yang dapat dideteksi lebih awal menggunakan teknologi *Structural Health Monitoring System (SHMS)*". Pada penelitian ini, teknologi GPS akan digunakan untuk memantau besarnya deformasi yang terjadi pada jembatan Suramadu (Maris, 2018). Pemanfaatan GPS digunakan untuk mengetahui seberapa besar pergerakan vertikal jembatan Suramadu. Hasil pengukuran yang diperoleh adalah pergerakan vertikal jembatan Suramadu sebesar $1,6385 \times 10^{-15}$ cm akibat adanya satu satuan mobil penumpang. Hasil pengukuran terhadap deformasi jembatan suramadu akan disimpan dan dimonitoring secara berkala. Pada penelitian tersebut hanya dilakukan pengukuran deformasi terhadap pergerakan vertikal jembatan saja.

Penelitian mengenai sistem monitoring jembatan juga telah dilakukan oleh Universitas Syiah Kuala pada tahun 2015. Penelitian tersebut membahas tentang penggunaan sensor akselerometer *MMA7361* sebagai alternatif pengukurann lendutan jembatan secara nirkabel berbasis atmega 32 (Syaryadhi, 2016). Penelitian yang lain mengenai sistem monitoring jembatan juga melakukan studi simulasi terhadap kesehatan jembatan dengan uji vibrasi. Penelitian tersebut membahas tentang metode pemeriksaan kerusakan jembatan rangka wren dengan *Non Destructive Testing* (NDT) berdasarkan uji vibrasi. Selain di Indonesia, penelitian tersebut juga di lakukan di negara Italia. Pada penelitian tersebut dilakukan pengujian keandalan jembatan beton bertulang terhadap struktural beban statis dan dinamis. Dalam uji beban statis, teknik leveling spirit presisi digunakan untuk mengukur defleksi geladak yang disebabkan oleh empat truk dengan berat masing-masing sekitar 36 ton (Gatti, 2019). Sensor akselerometer yang ditempatkan pada jembatan digunakan untuk mengukur frekuensi getaran. Negara jepang juga telah

menciptkan suatu penelitian mengenai sistem monitoring yang berjudul “*Research and Implementations of Structural Monitoring for Bridges and Buildings in Japan*”. Dalam penelitian tersebut menjelaskan tentang pengembangan sistem monitoring terhadap struktural jembatan dan bangunan. Sistem monitoring jembatan dan bangunan menggunakan teknik berbasis getaran. Hasil pengukuran deformasi terhadap jembatan dan bangunan tersebut disimpan pada database monitoring. Pada keempat penelitian tersebut hanya dilakukan sistem monitoring getaran yang disebabkan oleh deformasi terhadap perubahan beban baik untuk jembatan dan bangunan. Sedangkan sensor akselerometer hanya difungsikan untuk mengukur frekuensi getaran.

Selain sistem monitoring jembatan, penelitian mengenai sistem kontrol jembatan juga dilakukan di beberapa negara salah satunya di Korea. Penelitian tersebut menciptakan kontrol aktif getaran jembatan di bawah beban bergerak menggunakan *Tuned Mass Damper* (TMD) (Kwon, Kim, & Lee, 1998). Sistem kontrol jembatan yang diterapkan adalah dengan meletakkan *Tuned Mass Damper* (TMD) ditengah-tengah jembatan yang bertujuan untuk meredam getaran terhadap perubahan beban kendaraan. Respon jembatan di bawah kendaraan yang bergerak hanya dihitung dengan mempertimbangkan efek massa kendaraan. Penelitian ini juga mampu menjelaskan respon dinamis jembatan terhadap muatan bergerak. Pada penelitian yang lain juga mengimplementasi kontrol aktif pada jembatan *cable stayed*. Implementasi kontrol aktif pada jembatan *cable stayed* ini difokuskan pada kemampuan beradaptasinya jembatan terhadap dinamika efek yang dihasilkan oleh gempa bumi dan angin ekstrim (Girona & Aparicio, 2016). Selain itu, kontrol aktif juga digunakan untuk mengurangi kelelahan dalam kinerja sehari-hari jembatan yang disebabkan oleh beban kendaraan yang melwatinya (Girona & Aparicio, 2016). Sistem kontrol yang diterapkan pada kedua penelitian tersebut hanya dengan meletakkan *device* yang aktif dalam merespon perubahan beban jembatan. Kontrol aktif ini diterapkan untuk bereaksi terhadap pembebanan statis dengan mereduksi momen lentur yang tidak seimbang secara signifikan. Karena distribusi momen lentur dimodifikasi secara adaptif maka kelelahan pada jembatan dapat berkurang.

Dalam membangun sistem monitoring dan kontrol diperlukan algoritma yang mampu mengatasi permasalahan pada jembatan. Algoritma adalah kumpulan

instruksi yang dibuat secara jelas untuk menunjukkan langkah-langkah penyelesaian suatu masalah (Fletcher,1991). Beberapa algoritma diantaranya adalah algoritma *First In First Out* (FIFO) dan *Priority Scheduling*. *First In First Out* (FIFO) adalah sebuah metode pemecahan dalam masalah antrian yang dapat diterapkan dengan cara laporan pengaduan yang pertama kali masuk diasumsikan keluar pertama kali (Kusuma dewi, 2018). Sedangkan *Priority Scheduling* merupakan algoritma penjadwalan yang mendahulukan proses yang memiliki prioritas tertinggi (Kusuma dewi,2018). Penelitian mengenai algoritma jembatan sudah dilakukan di negara Irlandia. Algoritma yang digunakan adalah algoritma *moving force identification* (MFI) untuk mengukur berat kendaraan yang melintasi jembatan. Algoritma *moving force identification* (MFI) ini menggunakan teknik regularisasi tikhonov (Fitzgerald, Sevillano, Obrien, & Malekjafarian, 2017). Penggunaan regularisasi tikhonov ini memerlukan suatu metode baru agar dapat di aplikasikan pada *bridge weigh in motion*. Namun kelemahan pada algoritma ini adalah hanya bisa mengukur berat kendaraan roda dua saja.

Dalam mendapatkan informasi kondisi jembatan secara real time diperlukan *Internet of Things*(IoT) sebagai media penghubung antara *device* (sensor), jaringan internet dan server. Hal ini dapat memudahkan dalam mengakses informasi lebih cepat dan real time(Yu, Zhao, Shi, & Ou, 2013). Beberapa penelitian yang memanfaatkan *Internet of Things* (IoT) adalah penelitian yang berjudul “Sistem penilaian kondisi jembatan menggunakan respons dinamik dengan *Wireless Sensor Network* (WSN)”. Infrastruktur jembatan selalu dihadapkan dengan kondisi lingkungan sehingga diperlukan implementasi aplikasi WSN untuk mengawasi kondisi jembatan dan mengirimkan pesan waspada ke ruang kendali ketika terjadi kondisi yang tidak normal (Putra et al., 2018). Pengembangan sistem penilaian kondisi jembatan ini berdasarkan *natural frequency* berbasis getaran. Hasil eksperimen menunjukkan bahwa sistem dapat mengukur *natural frequency* dengan nilai yang mendekati hasil *Finite Element Analysis* (FEA). Penelitian lain yang berjudul “Sistem monitoring jembatan pada titik tertentu menggunakan *Internet of Things* (IoT)” juga telah dilakukan. Pada penelitian tersebut model dirancang menggunakan *Modul Linduino* yang terhubung ke ADC dan sensor laser (Pradana, 2017). Data hasil yang diperoleh akan dianalisis menggunakan algoritma median

filter. Modul GPRS yang terhubung dengan modul Arduino digunakan untuk mengirimkan data ke ubidot menggunakan ATCommand sehingga hasil transmisi data pada ubidot akan ditampilkan dalam bentuk grafik.

Di negara India juga telah melakukan penelitian yang berjudul “Design of Bridge Monitoring System based on Internet Of Things (IoT)”. Penelitian tersebut membahas tentang sistem pemantauan keselamatan jembatan berbasis Internet Of Things (IoT) dikembangkan menggunakan Teknologi WSN. Sistem ini dapat memonitor dan menganalisis secara real time kondisi jembatan dan lingkungannya, termasuk ketinggian air dan kondisi keselamatan lainnya (Sonawane, 2018). Selain di India, negara Jepang juga telah melakukan penelitian yang berjudul “*The Development and Field Evaluation of an IoT System of Low Power Vibration for Bridge Health Monitoring*”. Penelitian tersebut melakukan pengembangan sensor akselerasi nirkabel berdaya rendah dan penyebaran sensor pada gateway nirkabel berbasis Internet of Things (IoT) untuk pemantauan jembatan. Seluruh sistem divalidasi dalam uji lapangan di jembatan Chijing di Shanghai. Evaluasi lapangan menunjukkan bahwa sistem pemantauan jembatan IoT yang dikembangkan dapat fungsi akuisisi data real-time, transmisi, penyimpanan dan pemrosesan analitik untuk mensintesis informasi keselamatan jembatan (Tong, 2019). Dari beberapa penelitian tersebut hanya memanfaatkan *Internet of Things*(IoT) sebagai media untuk memberikan informasi mengenai kondisi jembatan berdasarkan *natural frequency* getaran pada titik-titik tertentu jembatan.

Dari beberapa penelitian diatas memaparkan bahwa sistem monitoring jembatan hanya digunakan untuk menganalisa getaran, sistem kontrol yang bekerja hanya berada pada device kontrol aktif yang bereaksi terhadap pembebanan statis sedangkan pemanfaatan *Internet of Things* belum ada diterapkan pada sistem monitoring dan kontrol jembatan. Maka dari itu diusulkan suatu sistem algoritma dimana ketika kendaraan yang memasuki jembatan agar tidak melebihi kapasitas beban maksimal jembatan. Kendaraan yang masuk jembatan telah di index oleh beberapa variabel dalam kurun waktu 24 jam. Beberapa variabel tersebut adalah waktu, identitas kendaraan berupa plat nomer, jenis kendaraan, berat kendaraan dan index. Namun dalam perancangan algoritma ini diperlukan asumsi dalam menentukan jumlah sensor akselerometer. Hal ini dikarenakan jumlah

akselerometer yang digunakan bergantung pada panjang jembatan dan dimensi panjang kendaraan terpanjang.

Berat kendaraan menjadi salah satu variabel fisis yang utama. Variabel fisis yang kedua adalah jumlah kendaraan yang masuk sebagai input dan kendaraan yang keluar adalah sebagai output. Variabel fisis yang ketiga adalah data kendaraan atau identitas kendaraan yaitu informasi plat nomer dan jenis kendaraan yang memanfaatkan CCTV. Algoritma yang diterapkan adalah algoritma yang diadopsi dari prinsip algoritma *First in First Out* (FIFO) . Dimana data yang pertama kali masuk akan dieksekusi terlebih dahulu. Ketika kendaraan yang masuk jembatan melebihi kapasitas beban jembatan maka portal akan menutup. *Internet of Things*(IoT) digunakan untuk mentransfer data kondisi jembatan melalui jaringan internet. *Internet of Things*(IoT) sebagai media penghubung antara *device*, jaringan internet dan server. Hal ini *Internet of Things*(IoT) dapat memudahkan dalam mengakses informasi lebih cepat dan real time.

1.2 Perumusan Masalah

Perumusan masalah dalam penelitian ini antara lain:

1. Bagaimana merancang algoritma kontrol jembatan agar kendaraan yang melewati tidak melebihi kapasitas jembatan?
2. Bagaimana memvalidasi perancangan algoritma pada sistem monitoring dan kontrol jembatan berbasis *Internet of Things* (IoT) secara *realtime* ?

1.3. Tujuan Penelitian

Tujuan dari penelitian ini berdasarkan perumusan masalah adalah sebagai berikut:

1. Melakukan perancangan algoritma kontrol agar kendaraan yang melewati tidak melebihi kapasitas jembatan.
2. Melakukan validasi perancangan algoritma pada sistem monitoring dan kontrol jembatan berbasis *Internet of Things* (IoT) secara *realtime*.

1.4 Batasan Masalah

Batasan masalah dalam penelitian ini diantaranya sebagai berikut:

1. Pengujian kebenaran algoritma menggunakan perangkat lunak *visual basic*.

2. Hasil rancangan algoritma di tampilkan dalam bentuk simulasi.
3. Data kendaraan berdasarkan data *excel* yang diasumsikan dapat mewakili data riil.
4. Jumlah akselerometer yang digunakan diasumsikan berdasarkan jarak kendaraan terpanjang dan jarak aman antar kendaraan.
5. Algoritma yang dirancang berpedoman terhadap prinsip kerja algoritma *First in First out*.
6. Data informasi kendaraan diakses melalui *website*.

1.5 Manfaat Penelitian

Penelitian perancangan algoritma untuk sistem monitoring dan kontrol jembatan berbasis *internet of things* (IoT) secara *real time* ini bermanfaat baik dalam segi ilmu maupun aplikasi. Penelitian ini diharapkan dapat membantu pemerintah dinas transportasi perhubungan khususnya transportasi darat sebagai salah satu solusi dalam memonitoring kondisi jembatan secara online dan *real time*. Selain itu penelitian ini dapat digunakan sebagai sistem manajemen dan perawatan jembatan agar jembatan bisa bertahan lebih lama.

(Halaman ini sengaja dikosongkan)

BAB 2

TINJAUAN PUSTAKA

2.1 Jembatan

Berdasarkan UU 38 Tahun 2004 bahwa jalan dan jembatan merupakan bagian dari sistem transportasi nasional yang mempunyai peranan penting terutama dalam hal mendukung bidang ekonomi, sosial dan budaya serta lingkungan. Jembatan secara umum adalah suatu konstruksi yang berfungsi untuk menghubungkan dua bagian jalan yang terputus oleh adanya rintangan-rintangan seperti lembah yang dalam, alur sungai, danau, saluran irigasi, kali, jalan kereta api, jalan raya yang melintang tidak sebidang dan lain-lain. Menurut Ir. H. J. Struyk dalam bukunya “Jembatan“, jembatan merupakan suatu konstruksi yang gunanya untuk meneruskan jalan melalui suatu rintangan yang berada lebih rendah. Jembatan menjadi salah satu jenis bangunan yang apabila akan dilakukan perubahan konstruksi maka tidak mudah untuk dapat dimodifikasi, biaya yang diperlukan relatif mahal dan berpengaruh pada kelancaran lalu lintas pada saat pelaksanaan pekerjaan. Jembatan diharapkan dapat bertahan kurang lebih 100 tahun untuk jembatan yang berukuran besar, sedangkan jembatan yang berukuran tidak terlalu besar dapat digunakan 50 tahun (Agustinus, 2017). Hal ini menunjukkan bahwa, disamping kekuatan dan kemampuan untuk melayani beban lalu lintas maka juga perlu diperhatikan bagaimana pemeliharaan jembatan yang baik.

Jenis jembatan berdasarkan fungsi, bahan konstruksi, lokasi dan tipe struktur telah mengalami perkembangan pesat sesuai dengan kemajuan jaman dan teknologi. Berdasarkan kegunaannya jembatan dapat dibedakan sebagai berikut (Rahadi, 2015):

- a. Jembatan jalan raya (*highway bridge*)
- b. Jembatan jalan kereta api (*railway bridge*)
- c. Jembatan jalan air (*waterway bridge*)
- d. Jembatan jalan pipa (*pipeway bridge*)
- e. Jembatan militer (*military bridge*)
- f. Jembatan pejalan kaki atau penyeberangan (*pedestrian bridge*).

Berdasarkan bahan konstruksinya, jembatan dapat dibedakan menjadi beberapa macam antara lain (Rahadi, 2015):

- a. Jembatan kayu (*log bridge*) merupakan jembatan yang terdiri dari bahan kayu dengan bentang yang relatif pendek.
- b. Jembatan beton (*concrete bridge*) merupakan jembatan yang konstruksinya terbuat dari material utama bersumber dari beton.
- c. Jembatan beton prategang (*prestressed concrete bridge*) merupakan jembatan dengan bahan berkekuatan tinggi.
- d. Jembatan baja (*steel bridge*) merupakan jembatan yang menggunakan berbagai macam komponen dan sistem struktur baja diantaranya *deck*, *girder*, rangka batang, pelengkung, penahan dan penggantung kabel.
- e. Jembatan komposit (*compossite bridge*) merupakan jembatan yang memiliki pelat lantai beton yang dihubungkan dengan *girder* atau gelagar baja yang saling menahan beban sebagai satu kesatuan balok.



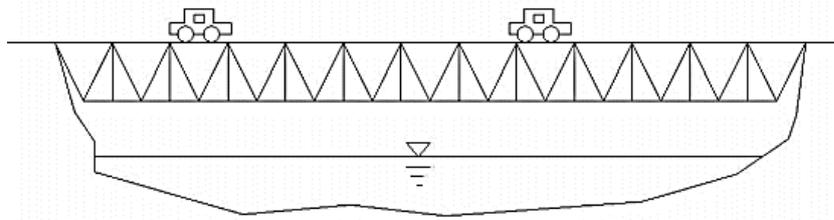
Gambar 2.1 Jembatan Suramadu (Direktorat Jembatan Bina Marga, 2016)

Deformasi pada jembatan suramadu, terdapat juga penelitian lain yaitu pengamatan lendutan vertikal jembatan kali babon dengan metode terrestrial laser scanner. Dalam Penelitian yang dilakukan oleh (Rizal Adhi Pratama, 2013) menggunakan Teknologi FARO *Laser Scanner Focus 3D* dalam memantau besarnya deformasi jangka pendek yang dihasilkan. Jumlah titik yang diamati yaitu sebanyak tujuh titik sepanjang bentang panjang jembatan. Kemudian dilakukan scanning sebanyak 11 kali agar memperoleh selisih lendutan maksimal. Hasil proses scanning berupa *point clouds* dalam bentuk tiga dimensi. Hasil pengukuran metode *terrestrial laser scanner* diperoleh lendutan maksimal sebesar 0.039 meter dengan lama waktu pengamatan selama 9 menit dalam sekali proses *scanning*. Rizal Adhi pratama (2013) menyimpulkan bahwa “Perhitungan teknis pembebanan

jembatan yang berubah-ubah berdasarkan *bridge management system* didapatkan nilai maksimal toleransi lendutan vertikal yang diperbolehkan adalah 0,083 meter". Hal ini menunjukkan bahwa lendutan Jembatan Kali Babon masih memenuhi standar (Rizal Adhi Pratama, 2013).

Klasifikasi jembatan berdasarkan letak lantai kendaraan dapat dibagi menjadi beberapa bagian, yaitu (Rahadi, 2015):

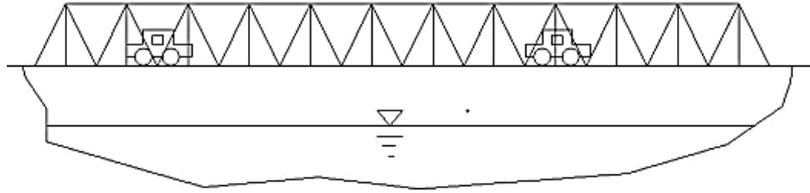
- a) Jembatan lantai bawah, dimana struktur rangka utama berada di atas lantai jembatan. Hal ini mengakibatkan batang bagian atas menjadi tertekan dan batang bagian bawah menjadi tertarik. Untuk batang bagian atas diperlukan pengaku untuk mengatasi bahaya tekuk. Biasanya pengaku ini berfungsi ganda karena dapat digunakan sebagai ikatan angin.



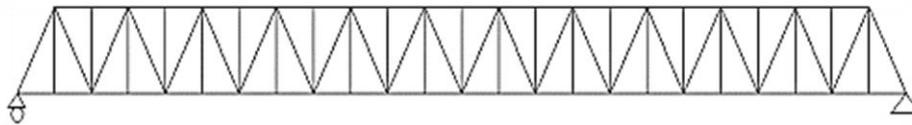
Gambar 2. 2 Jembatan lantai bawah (Rahadi, 2015)

Jembatan lantai bawah terbagi menjadi 2 yaitu jembatan rangka terbuka yaitu jembatan yang tidak memiliki ikatan angin dibagian atas. Jembatan ini cocok untuk lintas kendaraan yang berat karena bagian atas jembatan terbuka sehingga tidak menghalangi jalan. Jenis ini tidak memiliki ikatan angin dibagian atas. Jembatan ini cocok untuk lintas kendaraan yang berat karena bagian atas jembatan terbuka sehingga tidak menghalangi jalan untuk kendaraan berat. Sedangkan untuk jembatan rangka tertutup (dengan rangka atas). Jembatan rangka tertutup (dengan rangka atas). Jenis jembatan ini memiliki ikatan angin dibagian atas jembatan sehingga membentuk kotak (tertutup). Jenis jembatan ini cocok digunakan pada daerah perkotaan dan untuk lintas kendaraan yang ringan.

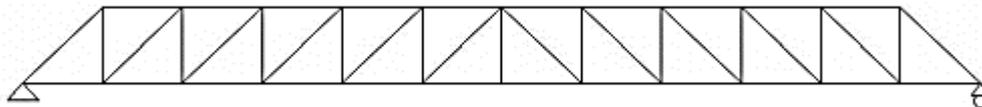
- b) Jembatan lantai atas, dimana struktur rangka jembatan ini berada dibawah deck jembatan. Jenis jembatan ini tidak cocok digunakan untuk sungai yang muka airnya rendah. Hal ini karena jenis jembatan ini memakan ruang yang ada dibawah lantai kendaraan.



Gambar 2. 3 Jembatan lantai Atas (Rahadi, 2015)



Gambar 2. 4 Jembatan rangka tipe Warren with verticals (Rahadi, 2015)



Gambar 2. 5 Jembatan rangka tipe *howe* (Rahadi, 2015)

- c) Konstruksi bangunan atas (*superstructure*)

Struktur atas jembatan merupakan bagian yang menerima beban langsung yang meliputi berat sendiri, beban mati, beban mati tambahan, beban lalu lintas kendaraan, gaya rem dan beban pejalan kaki. Struktur atas jembatan meliputi (Rahadi, 2015) :

- 1) Trotoar

Trotoar merupakan bagian layanan jembatan yang digunakan untuk sarana pejalan kaki, yang berada dibagian pinggir kiri dan kanan lantai kendaraan. Ketinggian trotoar lebih tinggi dari pada ketinggian permukaan lapisan lantai kendaraan. Trotoar terdiri dari Sandaran dan Slab lantai trotoar.

- 2) Lantai kendaraan dan perkerasan

Merupakan bagian konstruksi jembatan yang langsung menerima beban yang berjalan di atasnya. Di dalam perencanaan diperhitungkan terhadap beban hidup / muatan (T) dari tekanan roda kendaraan dan termasuk berat sendiri lantai kendaraan. Jika pelat beton dihubungkan pada balok memanjang dengan hubungan geser maka perhitungannya dapat menggunakan prinsip komposit.

Penelitian yang berhubungan dengan konstruksi jembatan adalah sebagai berikut "*Improvement of bridge structures to increase the safety of moving trains during earthquakes*". Menurut Ju (2013) berpendapat bahwa "Konstruksi desain jembatan yang dilewati oleh kereta yang bergerak selama gempa bumi ini menggunakan analisis elemen hingga". Jenis jembatan adalah jembatan multi bentang standar yang digunakan untuk sistem rel kecepatan tinggi. Sehingga Ju (2013) dapat menyimpulkan bahwa "Hasil studi parametrik dengan lebih dari 320 analisis menunjukkan bahwa peningkatan keselamatan kereta untuk jembatan multi bentang dengan beberapa kontinu rentang tidak dapat diamati". Dengan adanya dua gelagar dapat meningkatkan keamanan kereta saat bergerak. Selain itu, karena *natural frequency* kereta memiliki frekuensi rendah.

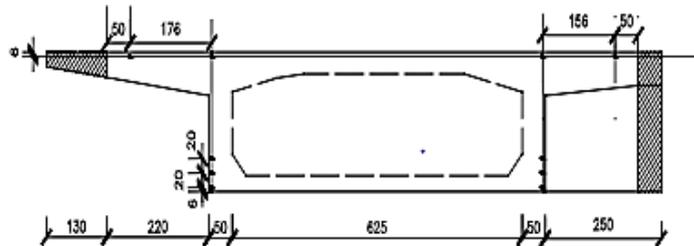
Salah satu penelitian berjudul "*Static Test Analysis of a Bridge Structure in Civil Engineering*" menjelaskan bagaimana dilakukan pengujian struktur terhadap jembatan. Melalui uji beban statis pada jembatan jalan dua jalur yang terletak di Pelabuhan Nanjiang dari Pelabuhan Tianjin. Konstruksi kualitas dan kinerja struktural jembatan dianalisis dan daya dukung jembatan yang sebenarnya ditentukan sehingga hal ini memberikan bukti ilmiah bahwa jembatan siap dioperasikan. Dilakukan *project overview* sebagai berikut (Zhao, Liu, & Wang, 2011) :

- 1) Project overviews.

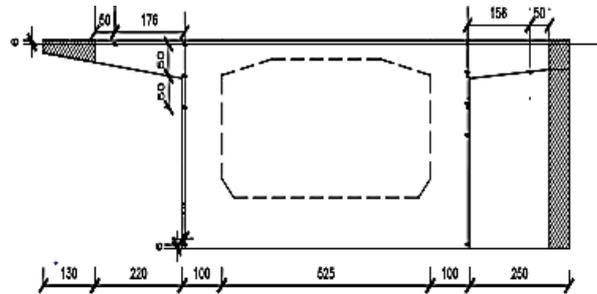
Mengamati Struktur jembatan yang akan di uji. Jembatan ini berada di pelabuhan nanjiang terletak di dekat pantai barat bohai. Jembatan utama digunakan girder kotak kontinu beton pratekan dengan penampang variabel 48,6m, 64m dan 48.6m, total lebar adalah 26,5 meter yang dibangun dengan dua arah secara terpisah. Tinggi girder adalah 4,0 meter titik tumpuan bentang utama, dan tinggi 2,0 meter di tengah bentang utama.



Gambar 2. 6 Struktur Jembatan(Zhao et al., 2011)



Gambar 2. 7 The sensors arrangement map of J2 section and J4 section(Zhao et al., 2011)

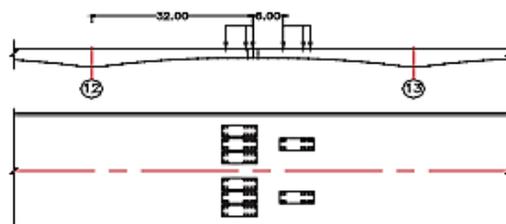


Gambar 2. 8 The sensors arrangement map of J2 section and J4 section(Zhao et al., 2011)

Pada gambar 2.6 dan gambar 2.7 menunjukan peletakan sensor untuk j1, j2 , j3 dan j4 menunjukkan bagian uji dan pengaturan titik pengukuran.

2) Dilakukan Uji Beban

Dilakukan uji beban dengan kendaraan berbobot 75Kn atau 76.435 ton dengan hasil uji seperti gambar 2.8 dibawah ini :



Gambar 2.7 The sensors arrangement map of J2 section and J4 section (Zhao et al., 2011)

Dalam penelitian ini, algoritma yang dibuat bisa diterapkan kepada jembatan-jembatan real sebagai contohnya adalah jembatan suramadu. Sehingga (Zhao et al., 2011) menjelaskan data karakteristik jembatan real berdasarkan konstruksi yang ada di jembatan berupa lebar jembatan dan jarak antar sisi sisi jembatan. Sehingga algoritma yang dibuat nantinya bisa diterapkan disetiap jembatan yang diuji. Kontribusi penelitian ini terletak pada pembuatan algoritma sistem jembatan yang safety terhadap kondisi jembatan, agar jembatan tidak terlalu cepat mengalami keruntuhan dan bisa memiliki umur yang panjang (Zhao et al., 2011).

2.2 Diagram Fishbone Penelitian

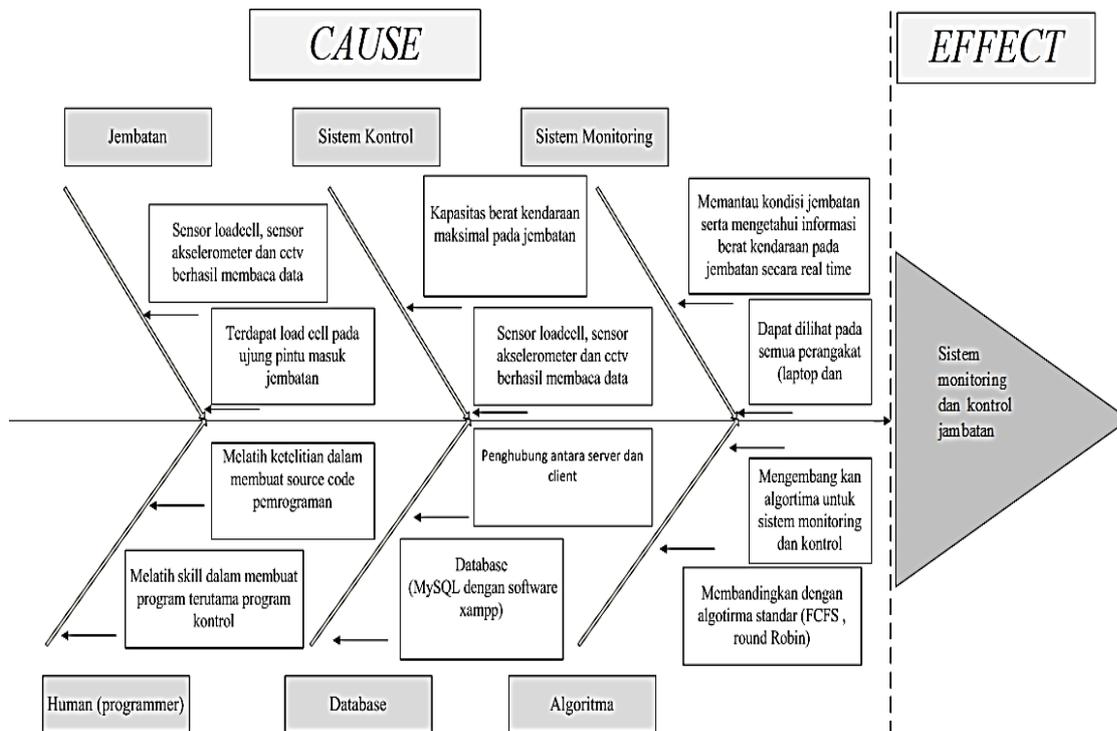
Diagram Cause and Effect atau Diagram Sebab Akibat adalah alat yang membantu mengidentifikasi, memilah, dan menampilkan berbagai penyebab yang mungkin dari suatu masalah atau karakteristik kualitas tertentu (Satya Wacana Christian, 2016). Diagram ini menggambarkan hubungan antara masalah dengan semua faktor penyebab yang mempengaruhi masalah tersebut. Jenis diagram ini kadang-kadang disebut diagram "Ishikawa" karena ditemukan oleh Kaoru Ishikawa, atau diagram "*fishbone*" atau "tulang ikan" karena tampak mirip dengan tulang ikan. Diagram fishbone ini dapat digunakan ketika kita perlu:

- Mengenal akar penyebab masalah atau sebab mendasar dari akibat, masalah, atau kondisi tertentu.
- Memilah dan menguraikan pengaruh timbal balik antara berbagai faktor yang mempengaruhi akibat atau proses tertentu.
- Menganalisa masalah yang ada sehingga tindakan yang tepat dapat diambil.

Manfaat menggunakan diagram fishbone ini:

- Membantu menentukan akar penyebab masalah dengan pendekatan yang terstruktur
- Mendorong kelompok untuk berpartisipasi dan memanfaatkan pengetahuan kelompok tentang proses yang dianalisis
- Menunjukkan penyebab yang mungkin dari variasi atau perbedaan yang terjadi dalam suatu proses.

Dalam penelitian ini dibuat diagram fishbone guna untuk mengenali akar penyebab masalah atau sebab mendasar dari akibat, masalah, atau kondisi tertentu. Serta Memilah dan menguraikan pengaruh timbal balik antara berbagai faktor yang mempengaruhi akibat yang ada pada sistem monitoring dan kontrol jembatan. Berikut diagram fishbone dari sistem monitoring dan kontrol jembatan:



Gambar 2. 9 Fishbone penelitian sistem monitoring dan kontrol jembatan

Dalam membuat sistem monitoring dan kontrol jembatan akan berjalan dengan baik apabila terdapat beberapa hal yang saling mendukung satu sama lain diantaranya sistem monitoring, sistem kontrol tidak berjalan, plant jembatan, Manusia, *Database* dan algoritma. Semua komponen yang terlibat bias saling mendukung, hal ini yang dapat mendukung sistem monitoring dan control dapat berjalan dengan baik. Untuk tiap faktor-faktor yang membuat keberhasilan sistem control dan monitoring adalah sebagai berikut:

1. Sistem Monitoring

Sistem monitoring dikatakan berhasil apabila sistem monitoring mampu memnatau kondisi jembata serta mengetahui informasi berat kendaraan pada jembatan secara real time. Selain itu siste monitoring dapat dilihat pada semua perangkat baik laptop, computer atau handphone. Namun beberapa potensi

kegagalan sistem monitoring bisa disebabkan oleh beberapa hal diantaranya data dari database tidak terkirim, internet lemot, data yang terkirim tidak sesuai dengan real plant dan real time sistem jembatan, *webservice error*. Guna untuk meminimalisir kegagalan sistem monitoring yaitu dengan mengecek koneksi internet dan mengoreksi apakah database yang ada sesuai atau tidak.

2. Sistem Kontrol

Sistem kontrol dikatakan berhasil apabila sistem kontrol mampu melakukan tindakan atau aksi apabila jembatan mengalami *overload* yaitu kelebihan muatan berat kendaraan. Selain itu sensor loadcell, akselerometer dan cctv berhasil membaca data dan terhubung dengan sistem kontrol jembatan berdasarkan algoritma. Namun potensi kegagalan sistem kontrol dapat terjadi banyak hal. Salah satu hal yang perlu diperhatikan yaitu input, proses dan output harus terkoneksi dan terhubung dengan baik. Sehingga potensi kegagalan dalam sistem kontrol bias diminimalisir. Berikut potensi kegagalan sistem kontrol diantaranya data yang dibaca oleh sensor *loadcell* gagal terbaca, kapasitas berat kendaraan maksimal belum ditentukan, tidak ada sinyal input dari sensor yang diterima oleh controller. Hal yang perlu diperhatikan dalam meminimalisir kegagalan sistem kontrol adalah dengan memastikan data data berat kendaraan sudah terbaca.

3. Algoritma

Algoritma adalah urutan langkah-langkah logis penyelesaian masalah yang disusun secara sistematis dan logis". Kata logis merupakan kata kunci dalam algoritma. Langkah-langkah dalam algoritma harus logis dan harus dapat ditentukan bernilai salah atau benar. Dalam beberapa konteks, algoritma adalah spesifikasi urutan langkah untuk melakukan pekerjaan tertentu. Pertimbangan dalam pemilihan algoritma adalah, pertama, algoritma haruslah benar. Artinya algoritma akan memberikan keluaran yang dikehendaki dari sejumlah masukan yang diberikan. Tidak peduli sebegus apapun algoritma, kalau memberikan keluaran yang salah, pastilah algoritma tersebut bukanlah algoritma yang baik.

Pertimbangan kedua yang harus diperhatikan adalah kita harus mengetahui seberapa baik hasil yang dicapai oleh algoritma tersebut. Hal ini penting terutama pada algoritma untuk menyelesaikan masalah yang memerlukan aproksimasi hasil

(hasil yang hanya berupa pendekatan). Algoritma yang baik harus mampu memberikan hasil yang sedekat mungkin dengan nilai yang sebenarnya.

Ketiga adalah efisiensi algoritma. Efisiensi algoritma dapat ditinjau dari 2 hal yaitu efisiensi waktu dan memori. Meskipun algoritma memberikan keluaran yang benar (paling mendekati), tetapi jika kita harus menunggu berjam-jam untuk mendapatkan keluarannya, algoritma tersebut biasanya tidak akan dipakai, setiap orang menginginkan keluaran yang cepat. Begitu juga dengan memori, semakin besar memori yang terpakai maka semakin buruklah algoritma tersebut. Dalam kenyataannya, setiap orang bisa membuat algoritma yang berbeda untuk menyelesaikan suatu permasalahan, walaupun terjadi perbedaan dalam menyusun algoritma, tentunya kita mengharapkan keluaran yang sama. Jika terjadi demikian, carilah algoritma yang paling efisien dan cepat.

Beberapa potensi kegagalan dalam membangun algoritma adalah konsep belum matang, alur atau langkah-langkah belum jelas bisa dikatakan masih rancu, algoritma yang dibangun tidak sesuai dengan konsep sistem yang dibuat.

4. Database

Dalam dunia database pasti pernah mengalami kegagalan dalam mengakses. Dan berikut ini adalah penyebab kegagalan pada database (Crash dan Recovery). Crash adalah suatu failure atau kegagalan dari suatu sistem. Beberapa penyebab kegagalan diantaranya. disk crash, informasi yang ada didisk akan hilang, Power failure, informasi yang disimpan pada memori utama dan register akan hilang. Selain itu *software error* yaitu output yang dihasilkan tidak betul dan sistem databasenya sendiri akan memasuki suatu kondisi tidak konsisten. Untuk klasifikasi kegagalan berdasarkan jenis storage adalah *volatile storage* dimana biasanya informasi yang terdapat pada volatile akan hilang, jika terjadi kerusakan sistem (system crash). Sedangkan untuk *non volatile storage*, biasanya informasi yang terdapat pada non volatile storage tidak akan hilang jika terjadi kerusakan sistem. Kemudian *stable storage* yaitu informasi yang terdapat dalam *stabel storage* tidak pernah hilang. Selain hal-hal diatas terdapat jenis kegagalan lain yaitu

- 1) Logical Error, program tidak dapat lagi dilaksanakan disebabkan oleh kesalahan input data tidak ditemukan.

- 2) System Error, sistem berada pada keadaan yang tidak diinginkan, seperti terjadi deadlock, sebagai akibat program tidak dapat dilanjutkan namun setelah beberapa selang waktu program dapat dijalankan kembali
- 3) System Crash, kegagalan fungsi perangkat keras, menyebabkan hilangnya data pada volatile storage, tetapi data pada non volatile storage masih tetap ada.

5. Manusia

Manusia menjadi faktor utama keberhasilan suatu project, pada kasus ini pembuat project harus bisa memprogram dalam bahasa komputer yang telah dipilih. Beberapa potensi kegagalan diantaranya mengalami kelelahan sehingga code yang dibuat salah, kesalahan dalam mendecclare dan menginialisasi dan tidak fokus menjadi potensi menjadi potensi kegagalan dalam membuat sistem pemrograman.

2.3 Sistem Monitoring dan Kontrol

Sistem monitoring kesehatan jembatan terutama pada konstruksi jembatan ini sangat penting agar dapat memberikan data struktur yang terukur, valid, dan terverifikasi secara berkala yang mampu menunjukkan gejala jembatan yang tak sehat. Maka dari itu sistem monitoring sangat penting untuk diterapkan karena serangkaian masalah yang mungkin terjadi pada struktur jembatan seperti runtuh, fondasi ambles, atau kegagalan struktur lainnya dapat dideteksi secara dini dan diantisipasi sebelum mencapai kritis. Sistem pemantauan ini sangat penting dikembangkan di Indonesia terlebih terdapat 22,85% jembatan yang berumur 35 tahun yang diharapkan jembatan mampu bertahan kurang lebih 100 tahun(Effendi, 2013). Sistem monitoring juga dapat menjadi teknologi yang dapat membuat masyarakat pengguna jalan tercipta keamanan penanganan yang tepat bagi struktur bangunan dalam masa layanan. Faktanya di Indonesia terdapat 1.050 kilometer atau 89.000 buah jembatan dan 332 kilometer atau 54.000 buah terletak di kabupaten/kota dan 660 kilometer atau 35.000 buah terletak di jalan nasional maupun provinsi(Effendi & Hendarto, 2014). Pada era globalisasi ini, sistem pemantauan jembatan yang simpel dan cepat sangat diperlukan untuk memantau kondisi jembatan dan meminimalisasi kegagalan konstruksi. Pasalnya tantangan jembatan saat ini yakni banjir, gerusan, beban yang berlebih maupun pergerakan tanah.

Pada penelitian sebelumnya dilakukan suatu monitoring kesehatan. Penelitian tersebut menjelaskan cara memonitoring jembatan dengan mendeteksi kerusakan yaitu dengan memasang sensor pada kendaraan yang lewat di jembatan dan mempertimbangkan kendaraan (Mei, Gül, & Boay, 2019). Teknik deteksi kerusakan yaitu menggunakan data yang dikumpulkan dari sensor yang dipasang pada sejumlah besar kendaraan yang lewat dikembangkan. Pertama, pendekatan berdasarkan koefisien cepstral frekuensi-mel (MFCC). Kemudian, versi yang ditingkatkan berdasarkan pada MFCC dan analisis komponen utama (PCA) yang memanfaatkan sensor seluler jaringan diusulkan untuk mengatasi kekurangan dalam pendekatan yang memanfaatkan tunggal pengukuran. Dalam pendekatan yang ditingkatkan, data percepatan pertama-tama dikumpulkan dari semua kendaraan dalam jangka waktu tertentu. Kemudian, fitur yang diubah yang terkait dengan kerusakan jembatan diekstraksi dari MFCC dan PCA. Kerusakan dapat diidentifikasi dengan membandingkan distribusi fitur yang diubah ini. Mei, Gül, & Boay (2019) menyimpulkan bahwa “Hasil dari numerik analisis dan percobaan laboratorium menunjukkan bahwa pendekatan tersebut tidak hanya mengidentifikasi keberadaan kerusakan, tetapi juga memberikan informasi yang berguna tentang tingkat keparahan”.

Sistem monitoring jembatan juga dilakukan oleh John Leander dengan judul “*Monitoring and enhanced fatigue evaluation of a steel railway bridge*”. Dalam penelitian tersebut Leander, Andersson & Karoumi (2010) menjelaskan inspeksi terhadap Jembatan Soderstrom di Stockholm pusat yang merupakan salah satu jembatan kereta api terpenting Swedia terdapat retakan di jaring balok baja utama. Temuan ini memprakarsai studi teoritis yang menunjukkan bahwa retakan berkembang terutama karena koneksi yang dirancang dengan buruk. Studi-studi juga menunjukkan hasil yang mengkhawatirkan tentang umur jembatan. Namun, tidak ada retakan atau kerusakan lainnya telah ditemukan pada komponen-komponen ini selama inspeksi. Untuk mengeksplorasi perbedaan antara indikasi teoretis dan realitas yang diinspeksi maka dibuat sistem monitoring jembatan dengan program pemantauan ekstensif dengan metode analisis mengenai sisa umur *fatigue jembatan* berdasarkan nilai-nilai teoritis dan terukur (Leander, Andersson, & Karoumi, 2010)

Selain di Indonesia, di Jepang juga telah menciptakan suatu penelitian yang mengangankan sistem monitoring yang berjudul “*Research and Implementations of Structural Monitoring for Bridges and Buildings in Japan*”. Dalam penelitian ini menjelaskan tentang pengembangan sistem monitoring struktural jembatan dan bangunan yang ada di Jepang. Sistem monitoring jembatan dan bangunan struktur tersebut menggunakan teknik berbasis getaran. Sistem pemantauan struktural di Jepang secara historis dimulai dengan tujuan mengevaluasi respons struktural terhadap peristiwa *ekstrem*. Dalam pengembangan sistem monitoring, data yang dikumpulkan tersebut digunakan untuk memverifikasi asumsi desain, memperbarui spesifikasi, dan memfasilitasi sistem kontrol getaran (Fujino, Siringoringo, Ikeda, Nagayama, & Mizutani, 2019). Strategi dan studi kasus pada sistem monitoring terdapat beberapa hal yang perlu diverifikasi yaitu desain jembatan, gedung-gedung tinggi, kinerja sistem isolasi seismik dalam bangunan dan jembatan, verifikasi retrofit struktural, verifikasi sistem kontrol struktural (pasif, semi-aktif, dan aktif), penilaian struktural, dan deteksi kerusakan. Baru-baru ini, aplikasi sistem monitoring telah diperluas untuk memfasilitasi operasi yang efisien dan pemeliharaan yang efektif melalui rasionalisasi risiko dan manajemen aset menggunakan data pemantauan (Fujino et al., 2019)

Selain itu sistem monitoring dan kontrol yang dilakukan oleh Yan Yu dengan judul “*Design of a real-time overload monitoring system for bridges and roads based on structural response*”. Yu (2013) menjelaskan tentang sistem monitoring dengan memantau struktural portabel, murah, berdaya rendah, yang dilengkapi dengan sensor lalu lintas *Polyvinylidene Fluoride (PVDF)*, *central processing unit (CPU)*, *sirkuit amplifier*, *sirkuit konverter Analog / Digital (A / D)*, *sirkuit komunikasi*, *sirkuit alarm*, dan perangkat lunak untuk menganalisis sistem proses. Pada sistem terdapat alarm yang digunakan sebagai pertanda bahwa jembatan dalam keadaan overload dimana data kelebihan beban kendaraan akan dikirim ke pusat untuk buat analisis lebih lanjut. Pada saat yang sama, nomor plat kendaraan yang kelebihan berat akan dikumpulkan dan dikirim ke departemen manajemen lalu lintas (Yu et al., 2013).

Sistem monitoring pada penelitian ini adalah sistem monitoring pada jembatan berbasis *internet of things (IoT)* secara real time. Data informasi dan berat

kendaraan yang melewati jembatan dapat diakses melalui website baik dibuka melalui computer, laptop maupun *handphone*. Informasi yang dapat di lihat pada website tersebut adalah data beban kendaraan, data informasi waktu kendaraan yang melewati dan informasi jembatan mengalami *overload* atau tidak. Tujuan dibuatnya sistem monitoring ini adalah agar pemerintah dapat memantau kondisi jembatan berdasarkan beban kendaraan yang melewati jembatan. Sehingga kondisi jembatan bisa terpantau dengan baik. Selain sistem monitoring, sistem kontrol jembatan juga sangat penting karena apabila kendaraan yang melewati jembatan dengan jumlah beban maksimal, maka sistem kontrol jembatan akan melakukan tindakan berupa tertutupnya portal yang berada di sisi pintu masuk jembatan untuk masing-masing arah.

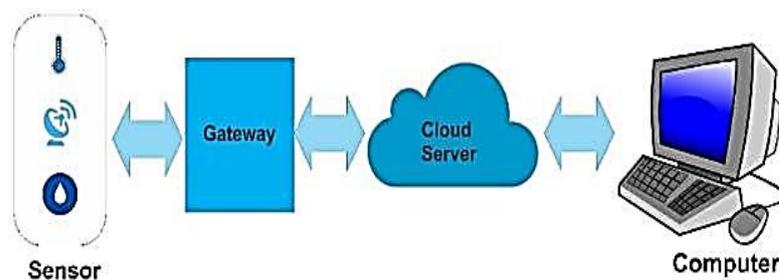
2.4 **Internet Of things(IoT)**

Internet of Things (IoT) pertama kali diperkenalkan oleh Kevin Ashton pada tahun 1999. Meski telah diperkenalkan sejak 15 tahun yang lalu, hingga kini belum ada sebuah konsensus *global* mengenai definisi IoT. Namun secara umum konsep IoT diartikan sebagai sebuah kemampuan untuk menghubungkan objek-objek cerdas dan memungkinkannya untuk berinteraksi dengan objek lain baik lingkungan maupun dengan peralatan komputasi cerdas lainnya melalui jaringan internet (Meutia, 2015). *Internet of Things* (IoT) dalam berbagai bentuknya telah mulai diaplikasikan pada banyak aspek kehidupan manusia. *Cisco Certification Network Associate* (CCNA) telah menargetkan bahwa “Pada tahun 2020 ada sekitar 50 miliar objek akan terhubung dengan internet”.



Gambar 2. 10 Internet of Things(Meutia, 2015)

Meluasnya adopsi berbagai teknologi *Internet Of Things*(IoT) dapat membuat kehidupan manusia menjadi jauh lebih nyaman. Apabila ditinjau dari sisi pengguna perorangan, *Internet Of Things*(IoT) sangat terasa pengaruhnya dalam bidang domestik contohnya pada aplikasi rumah dan mobil cerdas salah satu pengaplikasiannya yaitu dapat memungkinkan pemilik rumah dalam mengatur seluruh peralatan rumah dari jarak jauh dengan menggunakan satu aplikasi. Meutia (2015) juga mengatakan bahwa “*Internet Of Things* (IoT) juga yang diaplikasikan pada aplikasi peralatan medis seperti memonitoring glukosa yang terkoneksi pada pasien diabetes”. Hal tersebut akan memudahkan dokter menerima data pasien secara real time dalam memonitor kondisi pasien dan menyesuaikan dosis obat yang akan diberikan. Dengan demikian manajemen penyakit menjadi lebih mudah dilakukan (Meutia, 2015).



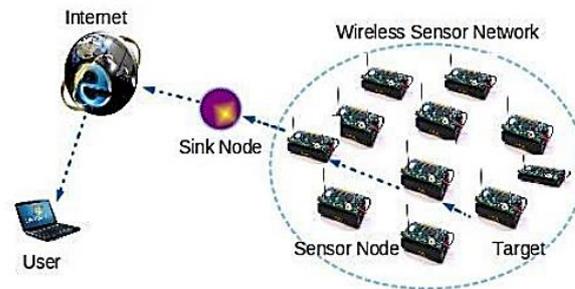
Gambar 2. 11 Blok Sistem *Internet Of Things*(Meutia, 2015)

Dalam mengimplementasikan *Internet of Things* (IoT) seperti pada contoh banyak teknologi yang terlibat antara lain: RFID sebagai alat pengenalan dan pengidentifikasi benda dan lokasi, teknologi web, WSN atau jaringan sensor nirkabel, dan Icloud. Teknologi-teknologi dalam IoT ini terhubung dengan berbagai terminal pengumpul data melalui jaringan internet maupun jaringan komunikasi lainnya. Informasi mengenai lingkungan di sekitar objek diambil secara *real time*, kemudian diubah ke dalam format data yang sesuai untuk ditransmisikan melalui jaringan kemudian dikirim ke pusat data (*Server*)(Yu et al., 2013).

2.5 *Wireless Sensor Network* (WSN)

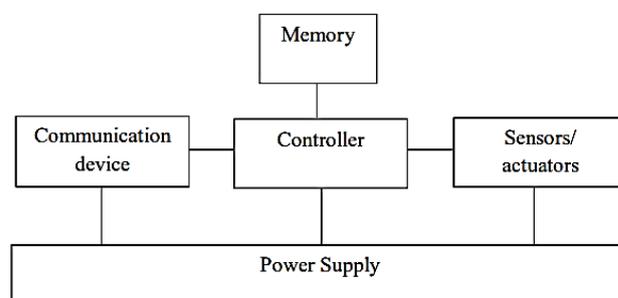
Wireless Sensor Network (WSN) atau jaringan sensor nirkabel merupakan kumpulan dari sejumlah node yang diatur dalam sebuah jaringan kerjasama (Moulat et al., 2018). Masing-masing node dalam jaringan sensor nirkabel pada umumnya

dilengkapi dengan radio *transceiver* atau alat komunikasi wireless lainnya, mikrokontroler dan *power supply* seperti baterai. Banyak aplikasi yang bisa dilakukan menggunakan jaringan sensor nirkabel seperti pengumpulan data, kondisi lingkungan, *security monitoring*, dan *node tracking scenarios* (Moulat et al., 2018).



Gambar 2. 12 Arsitektur WSN(Moulat et al., 2018)

Pada Gambar 2.12 menunjukkan gambaran umum dari *Wireless Sensor Network* (WSN) dapat dilihat node sensor tersebar dalam di suatu area sensor. Node sensor berfungsi untuk membuat rute data kemudian dikumpulkan ke node lain yang berdekatan. Data dikirimkan melalui transmisi radio akan diteruskan menuju BS (*Base Station*) atau *sink node* yang menjadi penghubung antara node sensor dan user (Li & Kara, 2017). Informasi tersebut dapat diakses melalui berbagai platform seperti koneksi internet atau satelit sehingga memungkinkan user untuk dapat mengakses secara *realtime*. Setiap node dalam *Wireless Sensor Network*(WSN) terdiri dari lima komponen yang saling terhubung diantaranya yaitu controller atau mikrokontroler, memori, sensor/akuator, perangkat komunikasi dan catu daya. Umumnya catu daya yang dipakai adalah baterai (Li & Kara, 2017).



Gambar 2. 13 Komponen yang terhubung dengan node (Li & Kara, 2017)

WSN menjadi kunci penting dalam memungkinkan terwujudnya *Internet Of Things*(IoT). Jenis serangan yang masuk dalam ketiga kategori ini adalah *denial of service* (DoS), yaitu serangan yang menyebabkan pengguna yang sah tidak dapat mengakses informasi (Adanur, Altunişik, Soyuk, Bayraktar, & Dumanoglu, 2016). Jaringan sensor nirkabel *Wireless Sensor Network* (WSN) merupakan perangkat tertanam kecil yang dipasang di jaringan skala besar. WSN mampu mengombinasikan sensor modern, mikroelektronika, komputasi, komunikasi, dan teknologi pemrosesan terdistribusi. Putra et al (2018) megusulkan penelitian mengenai pengembangan sistem pengukuran tingkat kesehatan struktur jembatan secara otomatis menggunakan respon-respon dinamis jembatan. Kontribusi utama dalam penelitian ini adalah pengembangan sistem penilaian kondisi jembatan berbasis getaran yang mengidentifikasi *natural frequency* dan bentuk getar jembatan. Hasil eksperimen menunjukkan bahwa sistem mampu mengukur *natural frequency* dengan nilai yang mendekati hasil finite element analysis (FEA) dan berdasarkan perhitungan Modal Assurance Criteriation (MAC), hasil pengukuran sistem yang dibangun memiliki korelasi tinggi dengan FEA (Putra et al., 2018).

2.6 Visual Basic Net(VBnet)

VB.NET adalah salah satu bahasa pemrograman Komputer Tingkat Tinggi. Salah Satu bahasa pemrograman yang *object oriented program*(OOP) atau Pemrograman yang berorientasi pada object pada kata "*Visual*" menunjukkan cara yang digunakan untuk membuat *Graphical User Interface* (GUI) dan tidak perlu lagi menuliskan instruksi pemrograman dalam kode-kode baris hanya untuk membuat sebuah *desaign form* hanya dengan cukup melakukan *drag and drop* objek-objek yang akan digunakan (Tylee, 1998). Bahasa pemrograman VB.NET dikembangkan oleh microsoft pemanfaatan aplikasi VB.NET diantaranya sistem aplikasi bisnis, software aplikasi sms, software aplikasi *chatting*, permainan. Dalam hal ini yang software vbnet akan menampilkan simulasi dari algoritma sistem jembatan, yang nantinya juga sebagai software yang menghubungkan dengan *Internet of Things (IoT)* (Tylee, 1998).

2.7 Algoritma

Algoritma memegang peranan penting dalam bidang pemrograman. Begitu pentingnya suatu algoritma, sehingga perlu dipahami konsep dasar algoritma.

Apalagi untuk seorang programmer, tentu saja diperlukan suatu algoritma sehingga dapat membuat program yang lebih efektif dan efisien. Bagi kebanyakan orang, algoritma sangat membantu dalam memahami konsep logika pemrograman. Algoritma adalah kumpulan instruksi yang dibuat secara jelas untuk menunjukkan langkah-langkah penyelesaian suatu masalah (Fletcher,1991). Pada umumnya algoritma kurang lebih sama dengan suatu prosedur yang sering dilakukan setiap hari, misalnya prosedur untuk mengganti ban bocor atau pecah, prosedur pemakaian telepon umum, prosedur membuat kue dan lain-lain. Guna memahami suatu algoritma, harus dimiliki pengetahuan dasar matematika karena pada dasarnya algoritma lahir dari konsep logika matematika. Disini yang perlu dilatih adalah kemampuan logikanya agar benar-benar bisa menyusun langkah-langkah penyelesaian masalah dengan baik. Algoritma adalah kumpulan instruksi/perintah yang dibuat secara jelas dan sistematis berdasarkan urutan yang logis (logika) untuk penyelesaian suatu masalah. French (1984) menyatakan bahwa “Sejumlah konsep yang mempunyai relevansi dengan masalah rancangan program yaitu kemampuan komputer, kesulitan dan ketepatan”. Penerapan dari konsep tersebut biasanya digunakan dalam rancangan algoritma. Dalam merancang sebuah algoritma, Fletcher (1991) memberikan beberapa cara atau metode yaitu kumpulan perintah, ekspresi, tabel instruksi, program komputer, kode semu dan flow chart, sedangkan Knuth (1973) menyarankan algoritma fundamental.

BAB 3

METODOLOGI PENELITIAN

3.1 Prosedur Penelitian

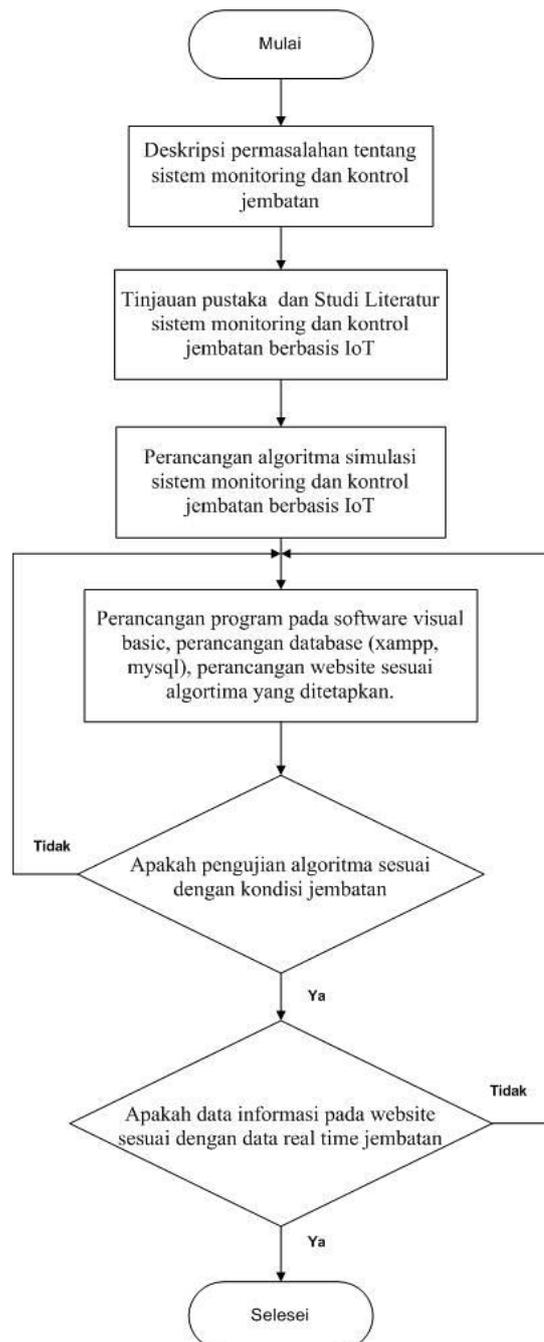
Prosedur dalam penelitian ini menerangkan kebutuhan metode perancangan algoritma dalam membangun sistem monitoring dan kontrol jembatan. Dalam merancang algoritma diperlukan gambaran mengenai kondisi jembatan tersebut. Secara umum kondisi-kondisi yang ada di jembatan diantaranya jembatan dalam kondisi macet dan lengang. Saat jembatan dalam kondisi macet dan kendaraan yang berada di jembatan adalah kendaraan besar maka beban yang diterima oleh jembatan akan besar dan sebaliknya. Dalam kondisi macet, kendaraan yang melewati jembatan pasti akan keluar dari jembatan sesuai dengan urutan ketika kendaraan memasuki jembatan. Sedangkan saat jembatan dalam keadaan lengang, kendaraan yang memasuki jembatan pada urutan pertama belum tentu keluar terlebih dahulu dari jembatan.

Beberapa variabel fisis yang berpengaruh diantaranya berat kendaraan yang melewati jembatan, variabel input output jembatan dan data kendaraan. Komponen yang dibutuhkan dalam perancangan sistem monitoring dan kontrol jembatan adalah sensor *load cell*, sensor akselerometer, *actuator* dan *controller*. Sensor *load cell* dirancang untuk mendeteksi tekanan atau berat kendaraan. Selama proses penimbangan akan mengakibatkan reaksi terhadap elemen logam pada *load cell* yang mengakibatkan gaya secara elastis. Gaya yang ditimbulkan oleh regangan ini dikonversikan kedalam sinyal elektrik oleh *strain gauge* yang terpasang pada *load cell*.

Kendaraan yang masuk sebagai input dan kendaraan yang keluar sebagai output. Pada saat kendaraan sudah memasuki jembatan dan sudah diukur beratnya oleh *load cell*, maka pada jembatan terdapat sensor akselerometer yang berfungsi untuk mendeteksi keberadaan kendaraan saat berada di jembatan. Data input yang diterima oleh jembatan adalah berupa data informasi mengenai plat nomer dan jenis kendaraan dengan menggunakan cctv. Sensor akselerometer dipasang pada setiap titik pada bentang jembatan. Peletakan sensor Akselerometer ditetapkan berdasarkan panjang jembatan dan panjang kendaraan terpanjang. Pada penelitian

ini jarak kendaraan satu dengan kendaraan lainnya diasumsikan 5 meter. Ketika kendaraan melewati sensor pertama data masih tersimpan, namun ketika kendaraan mengenai sensor kedelapan maka data yang tersimpan dalam *buffer* akan terhapus atau *ter-reset*. Data berat kendaraan yang masih tersimpan dalam *buffer* akan dijumlahkan. Apabila berat kendaraan yang berada di jembatan melebihi jumlah kapasitas maksimum maka portal akan menutup jembatan.

Secara umum diagram alir penelitian adalah sebagai berikut :



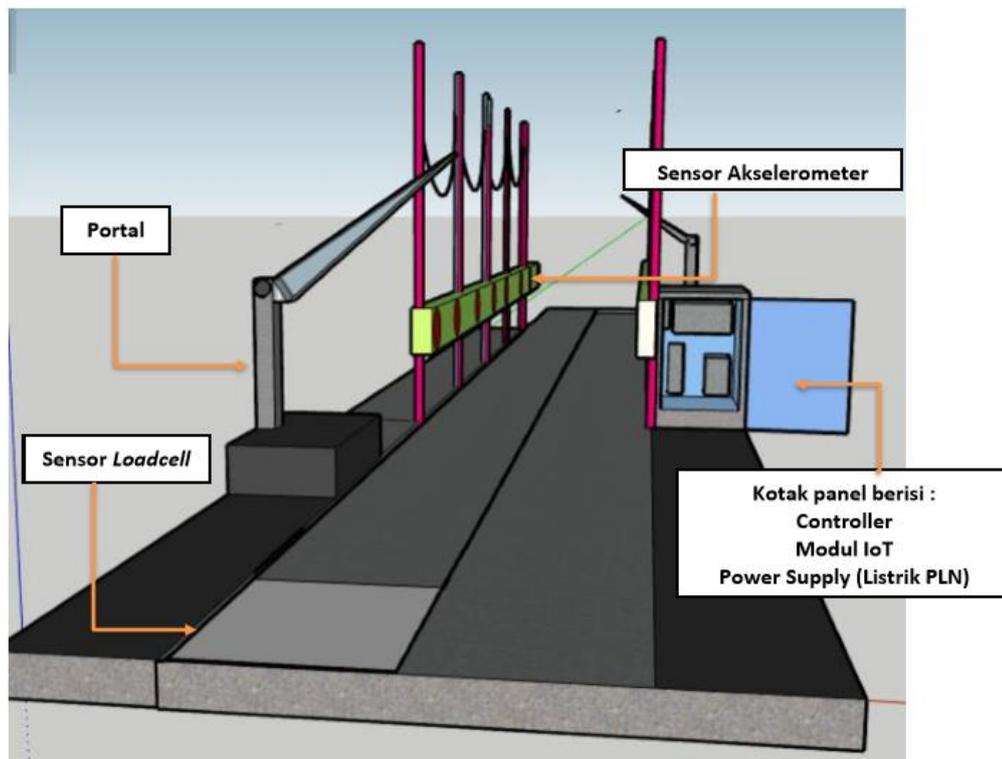
Gambar 3. 1 Diagram Alir Penelitian

Diagram alir penelitian pada gambar 3.1 diawali dengan mendeskripsikan permasalahan dan melakukan tinjauan pustaka tentang sistem monitoring dan kontrol jembatan. Langkah berikutnya adalah melakukan perancangan algoritma sistem monitoring dan kontrol jembatan berbasis *Internet of Things* (IoT). Untuk melihat kebenaran dari algoritma tersebut, dibuatlah program simulasi menggunakan *visual studio*, database menggunakan *mySql* dan aplikasi berbasis website.

3.2 Perancangan sistem

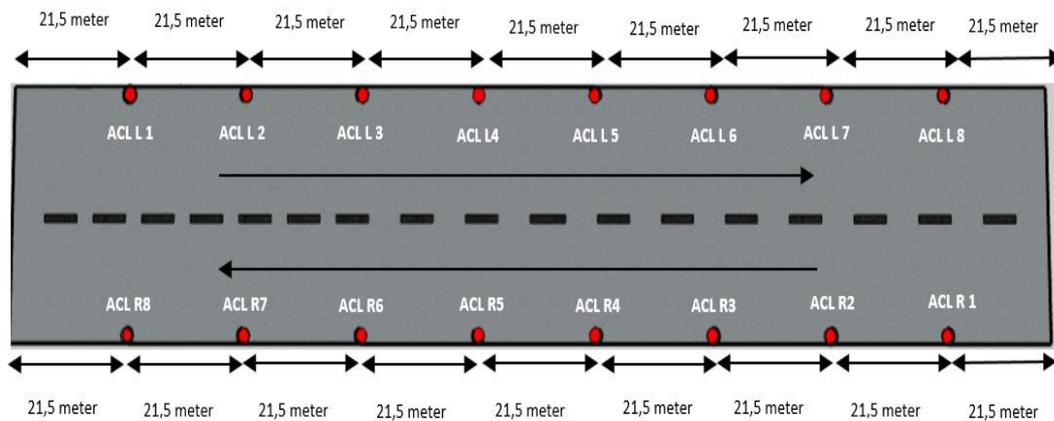
3.2.1 Perancangan Desain Jembatan

Sistem monitoring dan kontrol jembatan dibangun dari beberapa komponen, antara lain adalah sensor *loadcell* sebagai sensor berat kendaraan yang melewati jembatan, sensor *accelerometer* sebagai sensor untuk mendeteksi posisi kendaraan saat jembatan, portal sebagai aktuator pengaman jembatan saat berat total kendaraan melebihi kapasitas yang ditetapkan, komputer mikro sebagai pusat kendali, mikro SD sebagai penyimpanan data atau *logger* dan *power supply* dengan tegangan yang sesuai. Rancangan sistem monitoring dan kontrol jembatan ditampilkan pada gambar 3.1 sebagai berikut:



Gambar 3. 2 Desain jembatan

Dalam tesis ini diasumsikan bahwa panjang kendaraan maksimum 16,5 meter (Badan pusat statistika, 2011) dan jarak aman antar kendaraan adalah 5 meter. Peraturan mengenai jaga jarak antara kendaraan sudah diatur di dalam pasal 62 PP no. 43 tahun 1993 tentang tata cara berlalu lintas. Melalui peraturan tersebut mewajibkan para pengendara untuk menjaga jarak dengan mobil yang berada di depannya. Berdasarkan data dari Direktorat Jembatan Bina Marga pada tahun 2016 Indonesia tercatat memiliki 18.014 unit jembatan dengan panjang 481.926 meter. Maka dari itu pada penelitian ini diasumsikan panjang jembatan $\geq 193,5$ meter. Sketsa atau diagram rancangan penempatan sensor akselerometer ditampilkan pada gambar 3.2. Sedangkan untuk jarak peletakan sensor akselerometer ditampilkan pada gambar 3.3 sebagai berikut:



Gambar 3. 3 Desain peletakan sensor Akselerometer

Panjang kendaraan terpanjang di Indonesia adalah truk tronton *single maximum* dengan panjang 16,5 meter dan lebar 2,7 meter. Data tersebut didapatkan berdasarkan data badan pusat statistika untuk angkutan darat. Sedangkan jarak aman untuk kendaraan satu dengan yang lainnya adalah 5 meter berdasarkan aturan dari kepolisian tentang jarak aman. Maka asumsi perhitungannya adalah sebagai berikut :

Panjang jembatan = L

Jumlah Akselerometer = n

Panjang kendaraan terpanjang = x

Jarak kendaraan 1 dengan yang lainnya = y

Maka rumus yang digunakan adalah sebagai berikut :

$$L = ((n \cdot x) + (n-1) \cdot y) - 1 \dots \dots \dots (3.1)$$

Dengan diasumsikan data sebagai berikut:

$$L=193,5 \text{ meter}$$

$$x = 16,5 \text{ meter}$$

$$y = 5 \text{ meter}$$

Sehingga perhitungan untuk menentukan jumlah sensor Akselerometer yang dibutuhkan (n) adalah sebagai berikut:

$$L=((n \cdot x) +(n-1).y -)1)$$

$$193,5=((n.16,5)+(n-1)5)- 1)$$

$$193,5= (16,5n + (5n-5))-1$$

$$193,5= (16,5n+5n-5)-1$$

$$n= 8$$

Dari hasil perhitungan didapatkan jumlah sensor akselerometer yang digunakan adalah sebanyak 8 buah sensor dengan panjang jembatan yang ada adalah 167 meter. Namun apabila jembatan dengan panjang 102,5 meter maka perhitungan untuk jumlah sensor Akselerometer yang digunakan adalah sebagai berikut:

$$L=129 \text{ meter}$$

$$x = 16,5 \text{ meter}$$

$$y = 5 \text{ meter}$$

maka jumlah sensor Akselerometer yang dibutuhkan (n) adalah:

$$L=((n \cdot x) +(n-1).y)-1).....(3.2)$$

$$L=((n \cdot x) +(n-1).y -)1)$$

$$129=((n.16,5)+(n-1)5)- 1)$$

$$129= (16,5n + (5n-5))-1$$

$$129= (16,5n+5n-5)-1$$

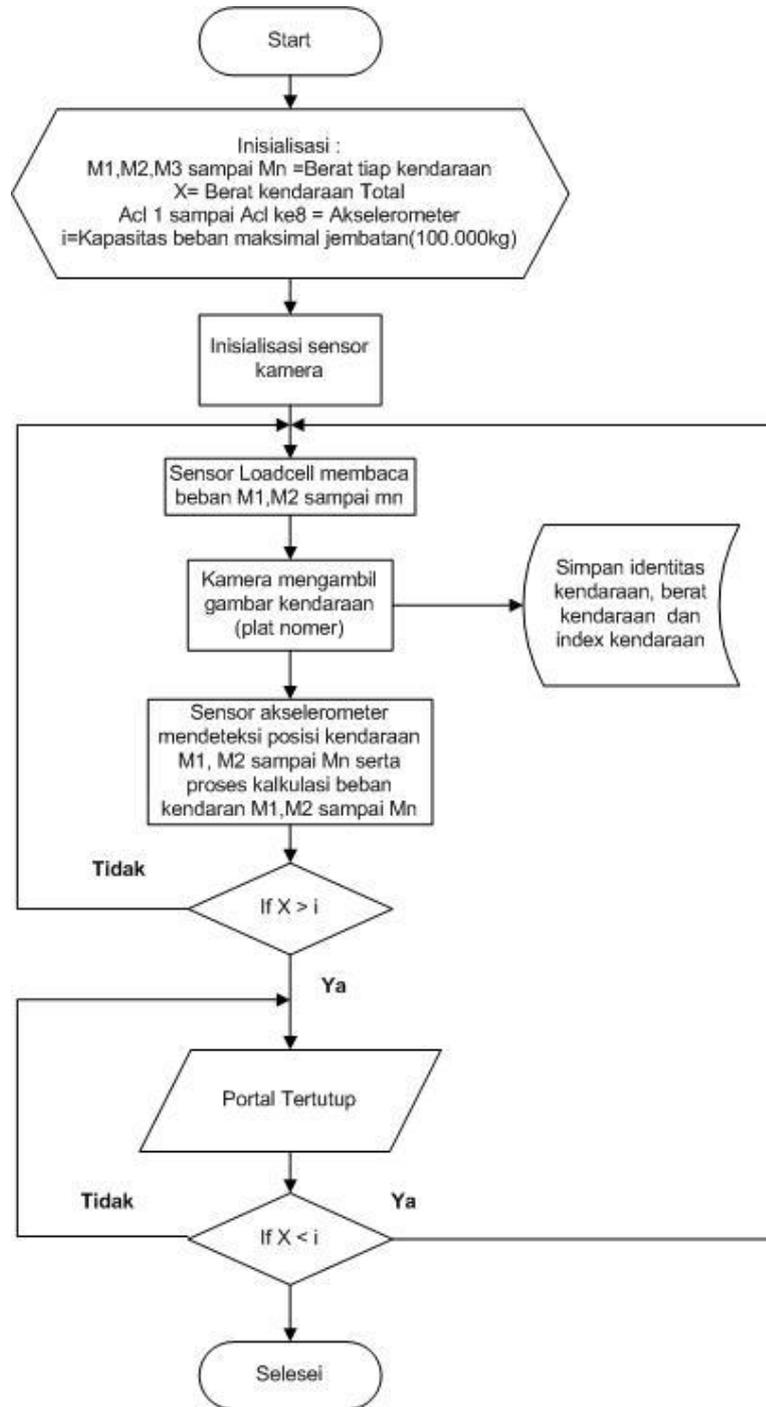
$$n= 5$$

Dari hasil perhitungan didapatkan jumlah sensor akselerometer yang digunakan adalah sebanyak 5 buah sensor dengan panjang jembatan yang ada adalah 129 meter.

3.2.2 Perancangan Algoritma dalam mengatur beban jembatan

Perancangan algoritma dalam sistem ini bertujuan untuk mengatur beban kendaraan yang melewati jembatan tidak melebihi kapasitas jembatan.

Berikut merupakan algoritma diagram alir dari kontrol beban pada jembatan :



Gambar 3. 4 Flowchart Algoritma Kontrol Jembatan

Perancangan algoritma kontrol jembatan ini diasumsikan bahwa kendaraan yang masuk jembatan di index dalam kurun waktu 24 jam. Beberapa variabel yang digunakan adalah sebagai berikut :

1. Waktu (hari dan tanggal)

Diperlukan informasi mengenai hari dan tanggal saat memasuki jembatan..

2. Identitas kendaraan (Plat Nomer)

Informasi mengenai identitas kendaraan saat memasuki jembatan yang berupa plat nomer juga disimpan bersamaan dengan penyimpanan waktu.

3. Jenis kendaraan

Jenis kendaraan bersifat optional artinya dapat digunakan untuk memberikan informasi mengenai jenis kendaraan.

4. Berat Kendaraan

Berat kendaraan dibaca oleh sensor loadcell yang berada di depan pintu masuk jembatan. Berat kendaraan tersebut akan di simpan dalam buffer.

5. Index

Kendaraan yang memasuki jembatan di index dalam kurun waktu 24 jam.

Tabel 3. 1 Ilustrasi proses algoritma kontrol jembatan berprinsip *First in first out* (FIFO)

	Proses Pembacaan Sensor		Proses Pendeteksian pada sensor Akselerometer								Keterangan	Total beban dalam buffer
	Sensor Laodcell	CCTV	A1	A2	A3	A4	A5	A6	A7	A8		
1	K1										•Sensor load cell mengukur berat kendaraan k1 , buffer menyimpan data berat kendaraan k1.	•Berat k1
2		K1									•Sensor kamera (cctv) mendeteksi identitas kendaraan k1 berupa plat nomor.	•Berat k1
3	K2		K1								•Sensor load cell mengukur berat kendaraan k2, buffer menyimpan data berat kendaraan k2. •Akselerometer 1 aktif, mendeteksi kendaraan k1	•Berat k1 + k2
4		K2		K1							•Akselerometer 2 aktif, mendeteksi kendaraan k1. •Sensor kamera (cctv) mendeteksi identitas kendaraan k2	•Berat k1 + k2
5	K3		K2		K1						•Akselerometer 3 aktif, mendeteksi kendaraan k1 •Akselerometer 1 aktif, mendeteksi kendaraan k2 •Sensor load cell mengukur berat kendaraan k2 data tersimpan	•Berat k1+k2+k3

Lanjutan tabel 3.1

	Proses Pembacaan Sensor		Proses Pendeteksian pada sensor Akselerometer								Keterangan	Total beban dalam buffer
	Sensor <i>Laodcell</i>	CCTV	A1	A2	A3	A4	A5	A6	A7	A8		
6		K3		K2	K1						<ul style="list-style-type: none"> •Akselerometer 3 aktif, mendeteksi kendaraan k1 •Akselerometer 2 aktif, mendeteksi kendaraan k2 •Sensor kamera (cctv) mendeteksi identitas kendaraan k3 berupa plat nomor k3 	•Berat k1+k2+k3
7	K4		K3		K2	K1					<ul style="list-style-type: none"> •Akselerometer 4 aktif, mendeteksi kendaraan k1. •Akselerometer 3 aktif, mendeteksi kendaraan k2 •Sensor kamera (cctv) mendeteksi identitas kendaraan k3 berupa plat nomor k3 •Sensor load cell mengukur berat kendaraan k4, buffer menyimpan data berat kendaraan k4. 	•Berat k1+k2+k3+k4
8		K4		K3		K2	K1				<ul style="list-style-type: none"> •Akselerometer 5 mendeteksi kendaraan k1 •Akselerometer 4 mendeteksi kendaraan k2, Akselerometer 2 mendeteksi k3 •Akselerometer 4 mendeteksi kendaraan k2, Akselerometer 2 mendeteksi k3 •Sensor kamera (cctv) mendeteksi identitas kendaraan k4 berupa plat nomor k4 	•Berat k1+k2+k3+k4
9			K4	K3			K2	K1			<ul style="list-style-type: none"> •Akselerometer 6 aktif, mendeteksi kendaraan k1. •Akselerometer 5 aktif, mendeteksi kendaraan k2 •Akselerometer 2 aktif, mendeteksi kendaraan k3 •Akselerometer 1 aktif, mendeteksi kendaraan k4 	•Berat k1+k2+k3+k4

Lanjutan tabel 3.1

	Proses Pembacaan Sensor		Proses Pendeteksian pada sensor Akselerometer								Keterangan	Total beban dalam buffer	
	Sensor <i>Laodcell</i>	CCTV	A1	A2	A3	A4	A5	A6	A7	A8			
10	K5			K4	K3				K2	K1	<ul style="list-style-type: none"> •Akselerometer 7 aktif, mendeteksi kendaraa1 •Akselerometer 6 aktif, mendeteksi kendaraan k2 •Akselerometer 3 mendeteksi kendaraan k3 •Akselerometer 2 mendeteksi k4 •Sensor load cell mengukur berat kendaraan k5, buffer menyimpan data berat kendaraan k5 	<ul style="list-style-type: none"> •Berat k1+k2+k3+k4+k5 	
11	K6	K5			K4	K3				K2	K1	<ul style="list-style-type: none"> •Akselerometer 8 aktif, mendeteksi kendaraan k1, berat k1 dihapus •Akselerometer 7 k2 aktif, mendeteksi kendaraan k2 •Akselerometer 4 k3 aktif, mendeteksi kendaraan k3 •Akselerometer 4 k3 aktif, mendeteksi kendaraan k3 •Sensor kamera (cctv) mendeteksi identitas kendaraan k5 berupa plat nomor k5 •Sensor load cell mengukur berat kendaraan k5, buffer menyimpan data berat kendaraan k5 	<ul style="list-style-type: none"> •Berat k2+k3+k4+k5+k6

Pada tabel 3.1 menunjukkan bahwa kondisi jembatan dalam keadaan macet, sehingga kendaraan 1 yang masuk pertama akan keluar pertama. Sehingga algoritma yang dibuat berprinsip algoritma *first in first out* (FIFO) atau FCFS (*first come first serve*) berlaku. Berikut algoritma yang diterapkan untuk menggambarkan kondisi tersebut:

1. Kendaraan masuk dari sisi kanan jembatan dan dari sisi kiri jembatan.
2. Sensor *loadcell* yang terletak disisi kanan dan kiri jembatan membaca berat kendaraan yang masuk dari sisi kanan jembatan dan dari sisi kiri jembatan.
3. Kamera yang terletak disisi kiri dan kanan jembatan mengambil gambar informasi kendaraan berupa plat nomer dan jenis kendaraan yang masuk dari sisi kanan jembatan dan dari sisi kiri jembatan.

4. Data berat kendaraan dari sisi kanan dan kiri jembatan disimpan pada *buffer*, sehingga kendaraan sudah terindex dengan data berat dan plat nomer.
5. Sensor Akselerometer mendeteksi keberadaan kendaraan yang berada pada kedua jalur jembatan.
6. Beban total kendaraan (n) = Beban total kendaraan (n-1) + beban kendaraan masuk dari sisi kiri+ beban kendaraan masuk dari sisi kanan.
7. Kendaraan kedua, kendaraan ketiga dan kendaraan ke-n memasuki jembatan dari sisi kanan dan sisi kiri , kembali ke langkah 2.
8. Kendaraan pertama, kendaraan kedua dan kendaraan ke-n melewati jembatan dengan berurutan sehingga sensor akselerometer pertama, akselerometer kedua dan akselerometer ketujuh mendeteksi kendaraan.
9. Kendaraan pertama, kendaraan kedua sampai kendaraan ke-n melewati akselerometer kedelapan (akselerometer terakhir yang berada diujung pintu keluar jembatan), maka data berat kendaraan tersebut dihapus.
10. Ketika kendaraan pertama, kendaraan kedua dan kendaraan ke-n belum terdeteksi oleh sensor kedelapan, maka berat kendaraan masih tersimpan di jembatan, kembali ke langkah 6.
11. Ketika beban total kendaraan (n) yang tersimpan dalam *buffer* kurang dari kapasitas beban maksimal jembatan maka portal jembatan masih terbuka, kembali ke langkah 1.
12. Ketika beban total kendaraan (n) yang tersimpan dalam *buffer* lebih dari kapasitas beban maksimal jembatan maka portal otomatis menutup jembatan.
13. Menunggu beberapa saat untuk terbuka sampai beban total kendaraan (n) kurang dari batas kapasitas maksimal, kembali ke langkah 1.
14. Data beban yang terdapat di *buffer* akan dikirim ke *webserver* dengan modul IoT secara periodik atau dalam waktu tertentu sehingga data yang dikirim sesuai dengan kondisi jembatan.

Tabel 3. 2 Ilustrasi proses algoritma kontrol pada kondisi lengang

	Proses Pembacaan Sensor		Proses Pendeteksian pada sensor Akselerometer								Keterangan	Total beban dalam buffer
	Sensor Laodcell	CCTV	A1	A2	A3	A4	A5	A6	A7	A8		
1	K1										<ul style="list-style-type: none"> •Sensor load cell mengukur berat kendaraan k1 , buffer menyimpan data berat kendaraan k1 	•Berat k1
2		K1									<ul style="list-style-type: none"> •Sensor kamera (cctv) mendeteksi identitas kendaraan k1 berupa plat nomor 	•Berat k1
3	K2		K1								<ul style="list-style-type: none"> •Sensor load cell mengukur berat kendaraan k2, buffer menyimpan data berat kendaraan k2 •Akselerometer 1 aktif, mendeteksi kendaraan k1 	•Berat k1 + k2
4		K2		K1							<ul style="list-style-type: none"> •Akselerometer 2 aktif, mendeteksi kendaraan k1 •Sensor kamera (cctv) mendeteksi identitas kendaraan k2 berupa plat nomor 	•Berat k1 + k2
5	K3		K2	K1							<ul style="list-style-type: none"> •Akselerometer 2 aktif, mendeteksi kendaraan k1 •Akselerometer 1 aktif, mendeteksi kendaraan k2 •Sensor load cell mengukur berat kendaraan k3, buffer menyimpan data berat kendaraan k3 	•Berat k1+k2 + k3
6		K3		K1	K2						<ul style="list-style-type: none"> •Akselerometer 3 aktif, mendeteksi kendaraan k2 •Akselerometer 2 aktif, mendeteksi kendaraan k1 •Sensor kamera (cctv) mendeteksi identitas kendaraan k3 berupa plat nomor k3 	•Berat K1+k2+ k3
7	K4		K3		K1	K2					<ul style="list-style-type: none"> •Akselerometer 4 aktif, mendeteksi kendaraan k2 •Akselerometer 3 aktif,mendeteksi kendaraan k1 •Akselerometer 1 aktif,mendeteksi kendaraan k3 	•Berat k1 + k2 +k3 + k4

Lanjutan tabel 3.2

	Proses Pembacaan Sensor		Proses Pendeteksian pada sensor Akselerometer								Keterangan	Total beban dalam buffer
	Sensor Laodcell	CCTV	A1	A2	A3	A4	A5	A6	A7	A8		
											<ul style="list-style-type: none"> • Sensor load cell mengukur dan menyimpan berat kendaraan k4. 	
8		K4			K3	K1	K2				<ul style="list-style-type: none"> • Akselerometer 5 aktif, mendeteksi kendaraan k2 • Akselerometer 4 aktif, mendeteksi kendaraan k1 • Sensor kamera (cctv) mendeteksi identitas kendaraan k4 berupa plat nomor k4 	<ul style="list-style-type: none"> • Berat k1+k2 + k3+k4
9	K5			K4		K3	K1	K2			<ul style="list-style-type: none"> • Akselerometer 6 aktif, mendeteksi kendaraan k2 • Akselerometer 5 aktif, mendeteksi kendaraan k1 • Akselerometer 3 aktif, mendeteksi kendaraan k3 • Akselerometer 2 aktif, mendeteksi kendaraan k4 • Sensor load cell mengukur berat kendaraan k5, buffer menyimpan data berat kendaraan k5 	<ul style="list-style-type: none"> • Berat k1+k2 + k3+k4+k5
10		K5				K4		K1	K3	K2	<ul style="list-style-type: none"> • Akselerometer 8 aktif, mendeteksi kendaraan k2, data kendaraan k2 terhapus. • Akselerometer 6 aktif, mendeteksi kendaraan k1 • Akselerometer 4 aktif, mendeteksi kendaraan k4 • Sensor kamera (cctv) mendeteksi identitas kendaraan k5 berupa plat nomor k5 	<ul style="list-style-type: none"> • Berat k2 terhapus • Berat k1+k3+ k4+k5
11	K6			K5				K4	K1	K3	<ul style="list-style-type: none"> • Akselerometer 8 aktif, mendeteksi kendaraan k3. • Akselerometer 7 aktif, mendeteksi kendaraan k1 • Akselerometer 6 aktif, mendeteksi kendaraan k4 • Akselerometer 2 aktif, mendeteksi kendaraan k5 • Sensor load cell membaca kendaraan 6 	<ul style="list-style-type: none"> • Berat k3 terhapus • Berat k1+k4+ k5+k6

Pada tabel 3.2 menunjukkan bahwa kondisi jembatan dalam keadaan lengang sehingga kendaraan pertama sampai kendaraan ke-n dapat saling mendahului.

Berikut algoritma untuk menggambarkan kondisi jembatan saat lengang:

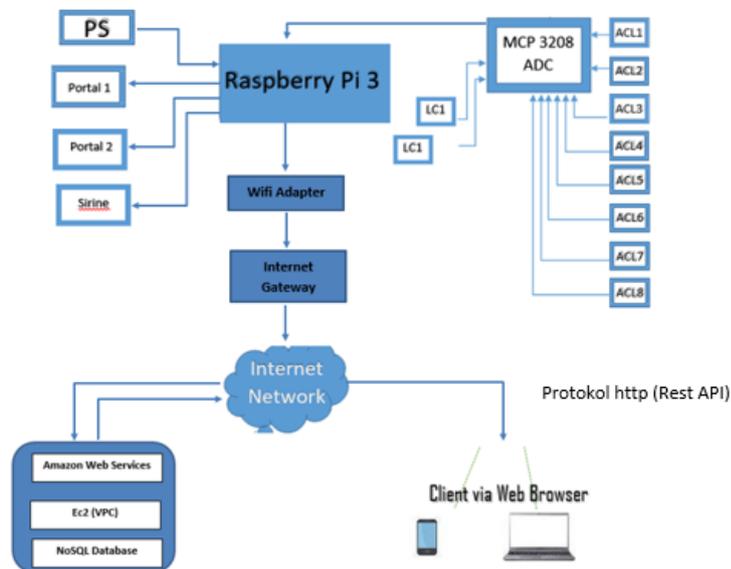
1. Kendaraan masuk dari sisi kanan jembatan dan dari sisi kiri jembatan.
2. Sensor *loadcell* yang terletak disisi kanan dan kiri jembatan membaca berat kendaraan yang masuk dari sisi kanan jembatan dan dari sisi kiri jembatan.
3. Kamera yang terletak disisi kiri dan kanan jembatan mengambil gambar informasi kendaraan berupa plat nomer dan jenis kendaraan yang masuk dari sisi kanan jembatan dan dari sisi kiri jembatan.
4. Data berat kendaraan dari sisi kanan dan kiri jembatan disimpan pada buffer, sehingga kendaraan sudah terindex dengan data berat dan plat nomer.
5. Sensor Akselerometer mendeteksi keberadaan kendaraan yang berada pada kedua jalur jembatan.
6. Beban total kendaraan (n) = Beban total kendaraan (n-1) + beban kendaraan masuk dari sisi kiri+ beban kendaraan masuk dari sisi kanan.
7. Kendaraan kedua, kendaraan ketiga dan kendaraan ke-n memasuki jembatan dari sisi kanan dan sisi kiri , kembali ke langkah 2.
8. Kendaraan pertama, kendaraan kedua dan kendaraan ke-n melewati jembatan dengan berurutan sehingga sensor akselerometer pertama, akselerometer kedua dan akselerometer ketujuh mendeteksi kendaraan.
9. Kendaraan pertama, kendaraan kedua sampai kendaraan ke-n melewati akselerometer kedelapan (akselerometer terakhir yang berada diujung pintu keluar jembatan), maka data berat kendaraan tersebut dihapus.
10. Ketika kendaraan pertama, kendaraan kedua dan kendaraan ke-n belum terdeteksi oleh sensor kedelapan, maka berat kendaraan masih tersimpan dijembatan, kembali ke langkah 6.
11. Apabila kendaraan kedua sampai kendaraan ke-n mendahului kendaraan (n-1) maka sensor akselerometer di sisi kanan atau kiri aktif mendeteksi kendaraan tersebut.
12. Apabila sensor akselerometer kedua sampai sensor ketujuh aktif bersamaan pada kedua sisi jembatan, maka berat kendaraan kendaraan kedua sampai

kendaraan ke-n akan ditambahkan pada loadcell berdasarkan kendaraan yang telah terindex, kembali ke langkah 1.

13. Ketika beban total kendaraan (n) yang tersimpan dalam *buffer* kurang dari kapasitas beban maksimal jembatan maka portal jembatan masih terbuka, kembali ke langkah 1.
14. Ketika beban total kendaraan (n) yang tersimpan dalam *buffer* lebih dari kapasitas beban maksimal jembatan maka portal otomatis menutup jembatan.
15. Menunggu beberapa saat untuk terbuka sampai beban total kendaraan (n) kurang dari batas kapasitas maksimal, kembali ke langkah 1.
16. Data beban yang terdapat di *buffer* akan dikirim ke *webserver* dengan modul IoT secara periodik atau dalam waktu tertentu sehingga data yang dikirim sesuai dengan kondisi jembatan.

3.2.3 Perancangan konsep sistem monitoring dan kontrol jembatan secara hardware

Pada perancangan konsep hardware ini membutuhkan beberapa komponen diantaranya sensor *loadcell*, sensor Akselerometer, *controller* dan modul IoT yang diperlukan. Untuk lebih jelasnya dapat dilihat pada gambar 3.6 sebagai berikut:



Gambar 3. 5 Arsitektur Hardware (Purbo, 2017)

Keterangan :

1. Ketika *loadcell* membaca berat kendaraan maka data berat tersimpan pada *buffer*. Semua total beban kendaraan yang melewati jembatandisimpan pada *buffer*. (MicroSD pada Raspberry pi 3).
2. Setiap data yang masuk akan dikalkulasi oleh Raspberry pi 3
(Kalkulasi artinya menghitung total beban dan total jml kendaraan)
3. Logic untuk safety Jembatan
Beban total kendaraan(n)= beban total kendaraan(n-1) + beban kendaraan dari sisi kanan + beban kendaraan dari sisi kiri
IF Beban total kendaraan (n) > 300.000 kg
Then portal menutup
Else
Portal tetap membuka
4. Logic Aliran data *Raspberry to Internet (Isp)*
Setiap data yang masuk ke Buffer (Micro SD) secara *real time* akan langsung di *direct* ke Web Server. Jika Link sudah sinkron dengan data Buffer (Micro SD) pada *raspberry* maka link buffer pada *web server* akan otomatis sama tampilan nilainya sesuai dengan data yang masuk pada Buffer (mikro sd). *Web server* bisa dibuka di perangkat mana saja (HP, Leptop, Tablet) dengan memasukan IP Address (Misal :172.17.2.64) atau domain name (misalnya : SMKJ.net) asal perangkat tersebut ada paket data atau koneksi internet. (Penentuan IP address atau domain name bisa diatur sesuai keinginan di dalam program Raspberry)
5. Web Service
Amazon Web Services adalah sekumpulan layanan-layanan berbasis *cloud computing* yang di sediakan oleh Amazon sejak tahun 2002. Database NoSQL menggunakan berbagai model data untuk mengakses dan mengelola data seperti dokumen, grafik, nilai kunci, dalam memori, dan pencarian. Jenis database ini dioptimalkan secara khusus untuk aplikasi yang memerlukan volume data besar, latensi rendah, dan model data fleksibel dengan mengurangi pembatasan konsistensi data dari database lainnya dibuat semacam *scheduler* pada *script web server* dimana setiap (missal : 5 menit) data pada *web server* akan *auto update*.

(artinya setiap 5 menit sekali *web server* di schedule mengambil data pada Raspb jadi tidak perlu tekan *refresh* atau *reload* pada browser).

Cara kerja dari masing-masing komponen apabila dalam bentuk hardware:

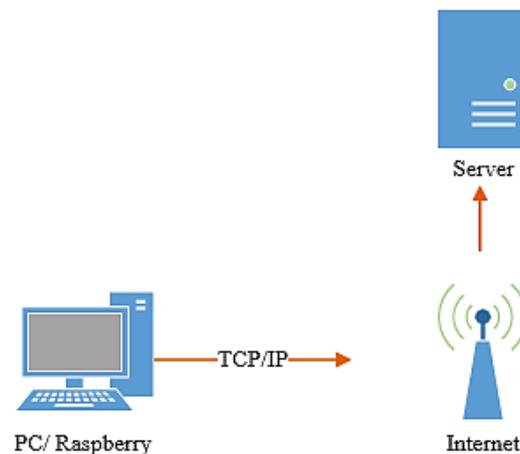
1. Proses sensor kamera membaca plat nomor kendaraan

- Pembacaan plat nomor menggunakan kamera yang kemudian diolah dengan pengolahan citra untuk mendapatkan karakter pada plat nomor.
- Banyak metode yang bias digunakan pada proses ini seperti *NN*, *CNN*, *deep learning*, dll. Untuk membantu pengolahan warna umumnya menggunakan bantuan *library OPEN-CV*.
- Untuk mempermudah pada umumnya dilapangan menggunakan modul pembaca plat nomer kendaraan dengan menggunakan kamera juga tentunya dengan berbagai merk yang beredar dipasaran.
- Data dari hasil pembacaan biasanya dapat diakses menggunakan protocol serial, modBus, 485, dll menyesuaikan dengan spesifikasi produknya.
- Dengan mengakses data hasil pembacaan dari modul melalui beberapa tipe protocol tersebut otomatis kerja dari controller (missal raspberry atau PC) utama akan lebih ringan, karena tidak terbebani dengan beratnya pengolahan citra yang dilakukan secara realtime pada gambar yang bergerak.

2. Proses sensor *loadcell* membaca beban kendaraan

- Pada process ini dibutuhkan modul loadcell itu sendiri, dengan kapasitas bisa disesuaikan.
- Serta amplifier untuk loadcell itu sendiri.
- Output amplifier untuk loadcell beragam, mulai dari ADC, Arus, oneWire dll.
- Lebih mudah jika menggunakan ADC namun akurasiya bergantung lebar pembacaan ADC pada controller.
- Controller seperti ARM umumnya mempunyai resolusi pembacaan ADC sampai 16 bit, sedangkan AVR rata rata 10 bit, sehingga akurasi untuk ketelitian pembacaan disarankan menggunakan ARM.

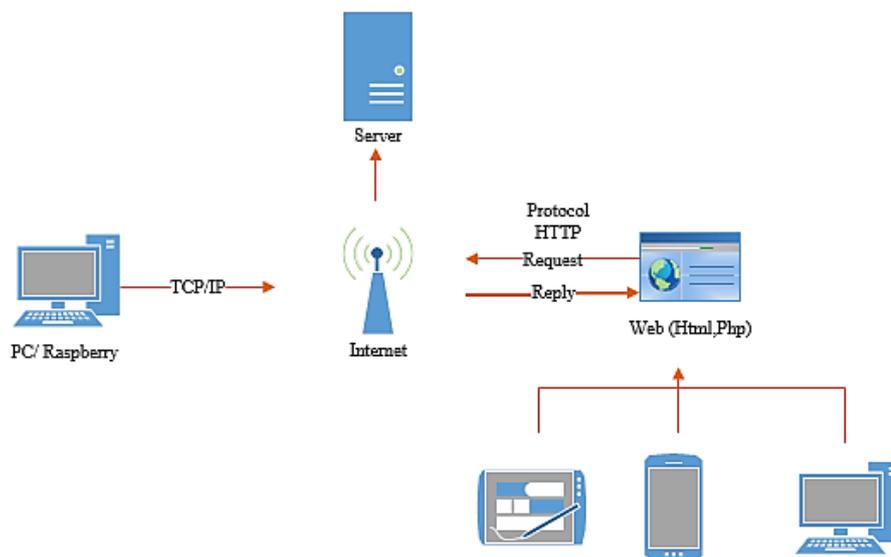
3. Pendeteksian kapasitas jembatan saat kendaraan masuk
 - Setelah diketahui beban kendaraan maka dijumlah dengan total beban yang sudah masuk jembatan dan sedang di atasnya.
 - Jika total beban jika ditambahkan dengan kendaraan yang akan masuk melebihi batas maksimum beban di atas jembatan maka portal tidak akan terbuka.
 - Namun jika total beban masih dalam kondisi aman maka portal dibuka dan kendaraan diizinkan lewat.
4. Proses penyimpanan data
 - Penyimpanan data dilakukan dengan oleh PC master atau bisa digantikan dengan raspberry.
 - Penyimpanan disesuaikan dengan format yang telah ditentukan.
 - Data disimpan pada PC server atau database online, seperti Xampp dll.
 - Proses penyimpanan membutuhkan akses internet sehingga dapat menjembatani antara PC master dengan server sehingga data dapat terhubung dan terupload pada database



Gambar 3. 6 Koneksi antara Rapsberry , internet dan server

5. Proses pendeteksian posisi kendaraan
 - Deteksi posisi kendaraan di atas kendaraan dideteksi dengan Accelero meter dengan jarak peletakan antar modul accelero yang telah ditentukan.
 - Pengiriman data posisi kendaraan ke PC master dapat dilakukan dengan menggunakan jaringan wireless, jika tidak dimungkinkan untuk menarik kabel komunikasi data sepanjang jembatan.

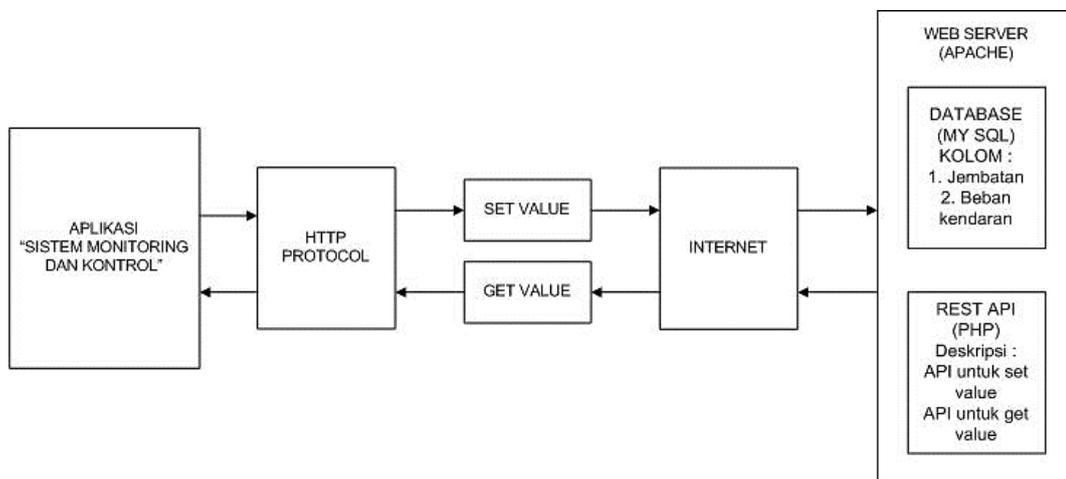
- Data posisi kendaraan akan menunjukkan posisi setiap kendaraan diatas jembatan.
 - Data posisi juga diupload pada database agar dapat disimulasikan, namun pada bagian ini hanya bersifat *current view* saja
6. Proses pengaksesan website
- Web pada bagian bertujuan untuk menampilkan data pada jembatan
 - Data yang ditampilkan merupakan data yang telah disimpan pada databse melalui Protocol HTTP.
 - Secara umum seluruh tampilan website dapat diakses atau ditampilkan pada device PC/Laptop, Hp, Tablet melalui web browser sehingga tidak perlu aplikasi penunjang.
 - Untuk perangkat client (PC/Laptop, Hp) agar dapat masuk pada access server atau database, pada umumnya hanya diberikan spesial akses melalui login akun yang kemudian akan terhubung dengan access server atau database melalui protocol HTTP.



Gambar 3. 7 Koneksi antar perangkat

3.2.4 Visualisasi alur konsep algoritma pada Simulasi Sistem Monitoring dan kontrol Jembatan

Pada perancangan simulasi ini dilakukan beberapa perancangan diantaranya program pada software visual studio atau visual basic, perancangan database menggunakan *software* xampp (database mySql) dan perancangan *website* yang nantinya menjadi media monitoring dalam mengamati kondisi jembatan secara *realtime*. Pada software *visual basic* telah dibuat algoritma sesuai dengan konsep dalam membangun sistem monitoring dan kontrol jembatan. Sedangkan data yang ada didalam *excel* berisikan data informasi kendaraan berupa berat kendaraan, plat nomer dan jenis kendaraan. Data yang berisikan informasi mengenai berupa berat kendaraan, plat nomer dan jenis kendaraan secara real tersimpan di dalam controller (raspberry). Software xampp berfungsi sebagai web server sedangkan MySQL berperan dalam mengolah, mengedit, dan menghapus daftar melalui database. Ketika data pada *excel* dibaca oleh *visual basic* maka data akan dikirim pada database MySQL. Proses update data berlangsung didalam database MySQL dan data pada database akan dikirim ke website berbasis IoT.



Gambar 3. 8 Protokol *Internet of Things* (IoT)

Perangkat *Internet of Things* (IoT) hanya menerima perintah melalui protokol komunikasi yang dikirimkan oleh sistem komputer. Protokol komunikasi bersifat agnostik sehingga sistem komputer dan IoT tidak harus mengembangkan protokol sendiri. Protokol ini bersifat agnostik sehingga dapat diimplementasikan pada penelitian ini. REST API adalah protokol komunikasi yang menggunakan endpoint URI dan memanfaatkan ketersediaan *method* pada Protokol HTTP, yaitu

Get Value untuk meminta data sedangkan *Set Value* adalah untuk membuat data baru. REST API memberikan layanan lengkap terhadap pengolahan data pada IoT dengan tetap bersifat agnostik.

A. Data Kendaraan

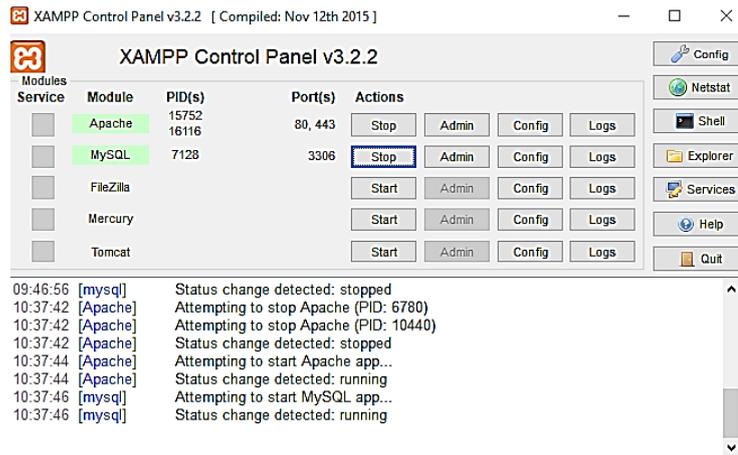
Data kendaraan yang terdapat pada lampiran merupakan data informasi yang terdiri dari jenis kendaraan, type kendaraan, plat nomer kendaraan dan berat kendaraan. Data tersebut menjadi nilai acuan dalam mengontrol beban jembatan dan juga sistem monitoring jembatan. Data kendaraan yang berisikan jenis kendaraan, type kendaraan, plat nomer kendaraan dan berat kendaraan akan disimpan dalam memori pada *controller raspberry*. Untuk mengetahui data dan informasi kendaraan dari lajur kanan dapat dilihat pada lampiran 1. Sedangkan untuk mengetahui data dan informasi kendaraan dari lajur kiri dapat dilihat pada lampiran 2.

B. Sistem Algoritma pada jembatan menggunakan software visual basic (Vb)

Software *visual basic* sebagai software yang membangun program algoritma. Hal-hal yang perlu diperhatikan adalah dengan memperhatikan penulisan kode program dengan memberikan indentasi yang konsisten di dalam kode program untuk membuat hubungan yang jelas antara kode blok, kelas dan fungsi. Program dari algoritma sistem kontrol dapat dilihat pada lampiran 3.

C. Database

Database yang digunakan adalah mySql pada software xampp control Panel sebagai web server. Pada software xampp bertindak sebagai web server terdiri atas Apache, MySQL, PHP dan Perl. Program ini tersedia dalam *General Public License* dan bebas yang merupakan web server yang mudah digunakan yang dapat melayani tampilan halaman web yang dinamis. Sebelum menyambungkan dengan sistem jembatan yang telah dibangun pada *visual basic* maka periksa dulu apakah xampp sudah beroperasi dengan baik atau tidak. Apabila xampp bisa digunakan dengan baik maka akan muncul tampilan seperti berikut ini :



Gambar 3. 9 Tampilan Xampp Control Panel

Pada gambar 3.9 menunjukkan bahwa xampp dapat terlink dengan sistem jembatan yang telah dibangun di visual basic ditandai dengan button apache dan MySQL sudah berwarna hijau. Apabila masih berwarna kuning atau merah maka perlu dilakukan perubahan settingan pada computer dengan mencari *service* kemudian *world wide web publishing* lalu *stop*. Setelah berwarna hijau, langkah selanjutnya adalah dengan memasukan data.sql kedalam table yang telah disediakan oleh xampp. Berikut tampilan data.sql yang berhasil dimasukan kedalam sistem.



Gambar 3. 10 Tampilan data MySQL

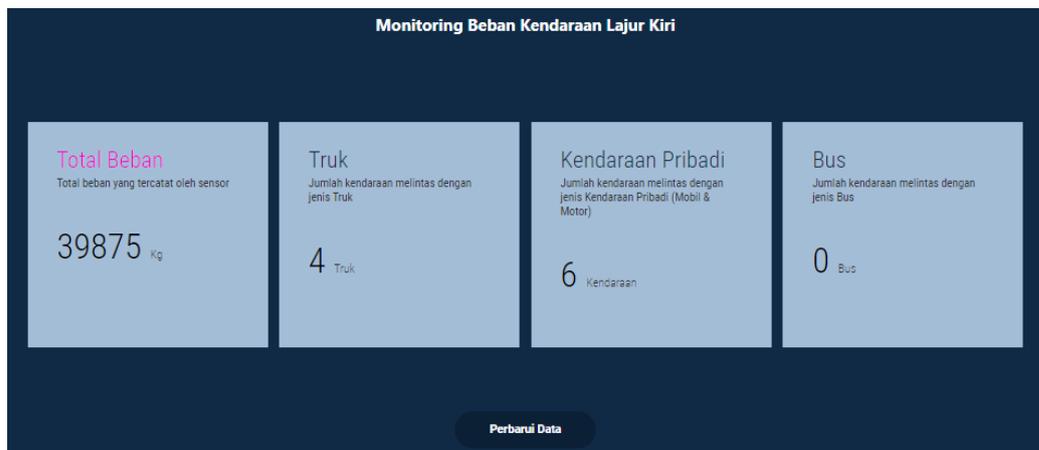
D. Website

Perancangan web adalah istilah umum yang digunakan untuk mencakup isi konten yang dikirimkan ke pengguna akhir melalui *world wide web* dengan menggunakan sebuah browser web atau perangkat lunak berbasis web. Tujuan dari

web design adalah untuk membuat website sekumpulan konten online termasuk dokumen dan aplikasi yang berada pada server. Sebuah website dapat berupa sekumpulan teks, gambar, suara dan konten lainnya, serta dapat bersifat interaktif ataupun statis. Untuk mengetahui program dari pembuatan website dapat dilihat pada lampiran 4. Desain tampilan website sistem monitoring dan kontrol jembatan berbasis *Internet of Things* (IoT) adalah sebagai berikut :



Gambar 3. 11 Tampilan Website



Gambar 3. 12 Tampilan Website pada monitoring kendaraan lajur kiri

Pada gambar 3.12 menampilkan data informasi berat total kendaraan yang berada di jembatan pada lajur kiri yaitu sebesar 39.875 kg. Selain data informasi berat kendaraan, pada website juga terdapat informasi jenis kendaraan yang melewati jembatan. Pada pengujian ini didapatkan informasi bahwa terdapat 4 truk, 6 kendaraan pribadi dan 0 bus.

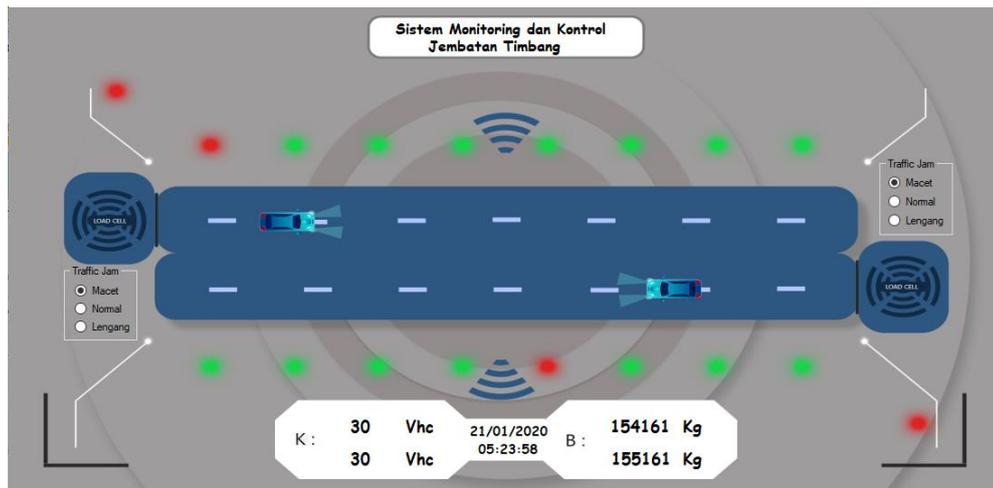


Gambar 3. 13 Tampilan Website pada monitoring kendaraan lajur kanan

Pada gambar 3.13 menampilkan data informasi berat total kendaraan yang berada di jembatan pada lajur kiri yaitu sebesar 86.865 kg. Selain data informasi berat kendaraan pada website juga terdapat informasi jenis kendaraan yang melewati jembatan. Pada pengujian ini didapatkan informasi bahwa terdapat 4 truk, 10 kendaraan pribadi dan 1 bus.

E. HMI Sistem Monitoring dan kontrol Jembatan

Pada desain HMI diatas terdapat sensor beban yaitu *load cell* yang ada pada setiap ujung jembatan saat mobil memasuki jembatan. Terdapat index K dan B yang artinya K adalah jumlah kendaraan dan B adalah beban yang diukur saat mobil memasuki jembatan. Sedangkan portal berada di setiap ujung pintu keluar jembatan untuk masing-masing jalur jembatan. Berikut desain HMI dari sistem jembatan :



Gambar 3. 14 Desain HMI pada simulasi sistem jembatan

Pada sepanjang bentang jembatan terdapat 16 sensor Akselerometer yang ditandai dengan warna merah dan hijau pada jembatan. Setiap jalur jembatan baik jalur kanan atau kiri terdapat masing-masing 8 sensor Akselerometer. Apabila kendaraan melewati sensor Akselerometer maka akan berubah menjadi warna hijau yang artinya sensor akselerometer aktif dalam mendeteksi posisi kendaraan. Namun setelah kendaraan melewati sensor *accleremoter 8* maka data kendaraan tersebut akan dihapus. Selain itu, pada tiap sisi jembatan terdapat portal yang akan menutup jembatan apabila beban kendaraan melebihi kapasitas beban jembatan.

F. Hubungan antara server, database dan client

Data yang dikumpulkan oleh perangkat *Internet of Things* (IoT) ini akan dikirim melalui jaringan internet dan LAN ke server untuk tujuan analisa dan menentukan output atau keputusan yang tepat sesuai data yang telah terkumpul. Dimana yang bertindak sebagai server adalah laptop yang sudah terinstall *software visual basic*. *Software visual basic* ini sudah terdapat program simulasi untuk sistem kontrol jembatan sesuai dengan algoritma yang ada.



Gambar 3. 15 Hubungan antara server, database dan client

MySQL berperan dalam mengolah, mengedit, dan menghapus daftar melalui database. Ketika data pada *excel* dibaca oleh *visual basic* maka data akan dikirim pada *database* MySQL. Proses update data berlangsung didalam database MySQL dan data pada database akan dikirim ke website berbasis (*Internet of Things*) IoT. Sedangkan *client* adalah perangkat yang digunakan untuk memonitoring informasi yang ada didalam server. Pada simulasi ini, ketika data informasi dari server dikirim ke client melalui database (*cloud*) dibutuhkan waktu beberapa detik agar data yang terkirim bisa terupdate. Database lainnya dibuat semacam *scheduler* pada *script web server* dimana setiap (missal: 5 menit) data pada web server akan auto update artinya setiap 5 menit sekali web server di jadwalkan mengambil data.

BAB 4

ANALISA DATA DAN PEMBAHASAN

4.1 Pengujian Algoritma

Algoritma yang dirancang perlu diuji kebenarannya sebelum diimplementasikan dalam sistem. Dalam tesis ini, pengujian kebenaran dilakukan dengan melakukan simulasi menggunakan perangkat lunak *visual basic*. Karakteristik program diusahakan mirip atau mendekati sama dengan kondisi sebenarnya, baik dalam jembatan, kendaraan yang melewati jembatan, sensor *loadcell*, sensor akselerometer maupun portal penutup jembatan. Data kendaraan yang akan melewati jembatan dipersiapkan dalam sebuah file *excel*. Data tersebut diasumsikan dapat mewakili data kendaraan riil.

4.1.1 Alur algoritma program pada masing-masing kondisi jembatan

Pengujian algoritma dilakukan dengan memberikan beberapa kondisi untuk jembatan. Sebelum melakukan pengujian diperlukan alur algoritma untuk masing-masing kondisi jembatan. Berikut adalah alur algoritma untuk masing-masing kondisi jembatan:

A. Alur algoritma program untuk kondisi jembatan lengang

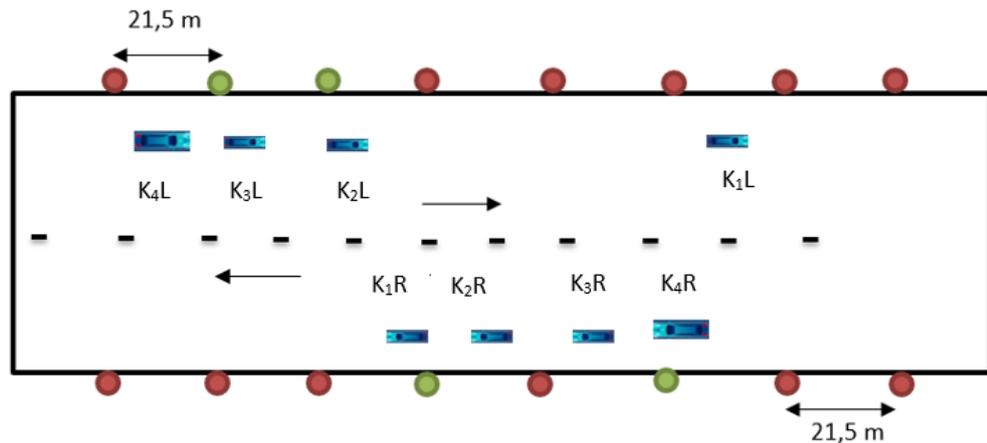
Kendaraan yang melewati jembatan dalam kondisi lengang memiliki kecepatan rata-rata 60 km/jam. Dalam kondisi ini, kendaraan yang memiliki kecepatan tinggi berpotensi mendahului kendaraan yang berada didepannya. Alur algoritma pada kondisi jembatan lengang adalah sebagai berikut:

1. Kendaraan(n) masuk pada lajur kiri atau kanan jembatan.
2. Sensor *loadcell* yang terletak disisi kanan dan kiri jembatan membaca berat kendaraan(n) yang masuk dari sisi kanan atau kiri jembatan.
3. Data beban kendaraan(n) yang masuk dari sisi kanan dan kiri jembatan tersimpan pada *buffer*.
4. Kendaraan(n) dari sisi kanan atau kiri jembatan terdeteksi oleh sensor akselerometer 1.
5. Beban total kendaraan (m) = Beban total kendaraan (m-1) + beban kendaraan yang masuk dari sisi kiri+ beban kendaraan kendaraan masuk dari sisi kanan.

6. Kendaraan(n+1) memasuki jembatan dari sisi kanan dan kiri, kembali ke langkah 2.
7. Kendaraan(n) dari sisi kanan dan kiri jembatan terdeteksi oleh sensor akselerometer 2 sedangkan kendaraan (n+1) dari kiri atau kanan diasumsikan memiliki kecepatan tinggi sebesar 60 km/jam sehingga kendaraan (n+1) terdeteksi oleh sensor akselerometer 3.
8. Kendaraan (n+2) baik dari sisi kanan dan kiri memasuki jembatan, kembali ke langkah 2.
9. Kendaraan (n+n) baik dari sisi kanan dan kiri memasuki jembatan, kembali ke langkah 2.
10. Kendaraan (n+1) terdeteksi oleh sensor akselerometer 4, kendaraan (n) terdeteksi oleh sensor akselerometer 3, kendaraan (n+2) terdeteksi oleh sensor akselerometer 2, kendaraan (n+n) terdeteksi oleh sensor akselerometer 1.
11. kendaraan (n+1) terdeteksi oleh sensor akselerometer 5, kendaraan (n) terdeteksi oleh sensor akselerometer 4, kendaraan (n+2) terdeteksi oleh sensor akselerometer 3, kendaraan (n+n) terdeteksi oleh sensor akselerometer 2.
12. kendaraan (n+1) terdeteksi oleh sensor akselerometer 6, kendaraan (n) terdeteksi oleh sensor akselerometer 5, kendaraan (n+2) terdeteksi oleh sensor akselerometer 4, kendaraan (n+n) terdeteksi oleh sensor akselerometer 3.
13. kendaraan (n+1) terdeteksi oleh sensor akselerometer 7, kendaraan (n) terdeteksi oleh sensor akselerometer 6, kendaraan (n+2) terdeteksi oleh sensor akselerometer 5, kendaraan (n+n) terdeteksi oleh sensor akselerometer 4.
14. kendaraan (n+1) terdeteksi oleh sensor akselerometer 7, kendaraan (n) terdeteksi oleh sensor akselerometer 6, kendaraan (n+2) terdeteksi oleh sensor akselerometer 5, kendaraan (n+n) terdeteksi oleh sensor akselerometer
15. Kendaraan (n+1) baik dari sisi kanan dan kiri jembatan terdeteksi oleh sensor akselerometer 8, sehingga data kendaraan(n+1) terhapus, kembali ke langkah 5.

16. Apabila beban total kendaraan (m) melebihi kapasitas beban maksimal jembatan maka portal tertutup, menunggu hingga beban total kendaraan (m) tidak melebihi kapasitas beban maksimal.
17. Apabila beban total kendaraan (m) tidak melebihi kapasitas beban maksimal jembatan maka portal terbuka, kembali ke langkah 1.
18. Data beban total kendaraan (m) dikirim ke website sesuai dengan kondisi jembatan real time.

Pada simulasi visual basic terdapat *Human Machine Interface* (HMI) yang menunjukkan gambaran ilustrasi simulasi dari sistem monitoring dan kontrol jembatan. Sensor akselerometer yang berada di sepanjang jembatan difungsikan untuk mendeteksi posisi kendaraan. Dapat dilihat pada gambar 4.1 menunjukkan bahwa apabila sensor mendeteksi kendaraan maka sensor akan berwarna merah dan sebaliknya.



Gambar 4. 1 Jembatan dalam kondisi lengang

Pada gambar 4.1 menunjukkan bahwa kondisi jembatan dalam kondisi lengang. Jumlah kendaraan yang ada di jembatan tidak terlalu banyak sehingga kendaraan satu dengan kendaraan yang lainnya bisa saling mendahului. Pada gambar 4.1 terlihat bahwa jumlah kendaraan yang ada di jembatan sebanyak 4 kendaraan untuk lajur kiri dan 4 kendaraan untuk lajur kanan. K_{4R} dan K_{4L} sebagai kendaraan terpanjang sedangkan K_{1L}, K_{2L}, K_{3L}, K_{1R}, K_{2R}, K_{3R} sebagai kendaraan dengan ukuran sedang dan sensor yang aktif 2 dan bisa lebih bergantung posisi kendaraan saat berada di jembatan. Apabila kondisi jembatan macet maka jenis K_{4R} dan K_{4L} hanya bisa 8 kendaraan dalam tiap jalur jembatan. Hal ini disebabkan panjang kendaraan

terpanjang adalah 16,5 meter dan jarak aman adalah 5 meter. Hal ini yang menyebabkan jarak peletakan sensor akselerometer adalah 21,5 meter.

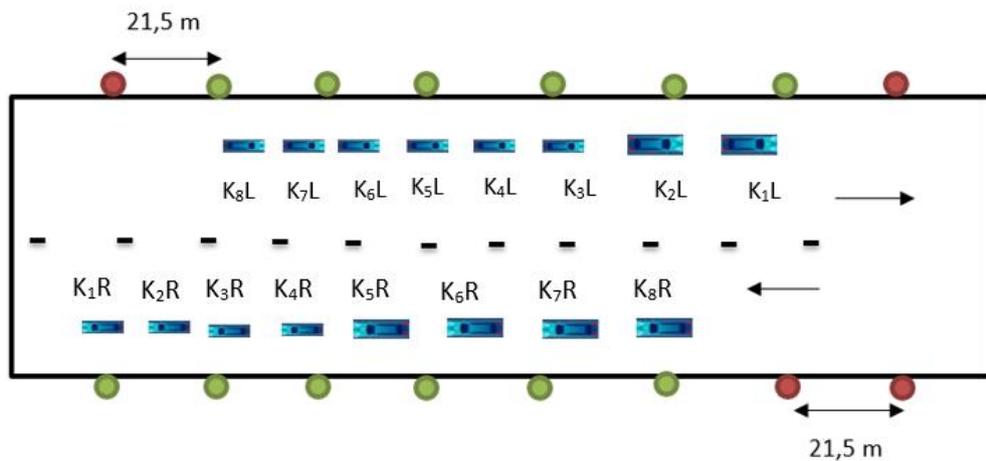
B. Algoritma untuk kondisi jembatan macet

Kendaraan yang melewati jembatan dalam kondisi lengang memiliki kecepatan rata-rata 5 km/jam. Dalam kondisi ini, kendaraan tidak dapat mendahului kendaraan yang berada didepannya. Alur algoritma pada kondisi jembatan macet adalah sebagai berikut:

1. Kendaraan(n) masuk pada lajur kiri atau kanan jembatan
2. Sensor *loadcell* yang terletak disisi kanan dan kiri jembatan membaca berat kendaraan(n) yang masuk dari sisi kanan atau kiri jembatan.
3. Data beban kendaraan(n) yang masuk dari sisi kanan atau kiri jembatan tersimpan pada *buffer*.
4. Kendaraan(n) dari sisi kanan atau kiri jembatan terdeteksi oleh sensor akselerometer 1.
5. Beban total kendaraan (m) = Beban total kendaraan (m-1) + beban kendaraan yang masuk dari sisi kiri+ beban kendaraan kendaraan masuk dari sisi kanan.
6. Kendaraan (n+1) memasuki jembatan dari sisi kanan dan kiri, kembali ke langkah 2.
7. kendaraan(n) dari sisi kanan atau kiri jembatan terdeteksi oleh sensor akselerometer 2.
8. kendaraan (n+1) terdeteksi oleh sensor akselerometer 3.
9. kendaraan (n+2) memasuki jembatan dari sisi kiri atau kanan jembatan, kembali ke langkah 2.
10. Kendaraan (n+n) memasuki jembatan dari sisi kiri atau kanan jembatan, kembali ke langkah 2.
11. Kendaraan (n+1), (n+2) dan (n+n) tidak dapat mendahului kendaraan (n).
12. Kendaraan (n), (n+1), (n+2) dan (n+n) memiliki kecepatan kurang lebih 5km/jam karena kondisi jembatan sedang padat, kembali ke langkah 1.

13. Apabila beban total kendaraan (m) melebihi kapasitas beban maksimal jembatan maka portal tertutup, menunggu hingga beban total kendaraan (m) tidak melebihi kapasitas beban maksimal.
14. Apabila beban total kendaraan (m) tidak melebihi kapasitas beban maksimal jembatan maka portal terbuka, kembali ke langkah 1.
15. Data beban total kendaraan (m) dikirim ke website sesuai dengan kondisi jembatan real time.

Pada simulasi visual basic terdapat *Human Machine Interface* (HMI) yang menunjukkan gambaran ilustrasi simulasi dari sistem monitoring dan kontrol jembatan. Sensor akselerometer yang berada di sepanjang jembatan difungsikan untuk mendeteksi posisi kendaraan. Dapat dilihat pada gambar 4.2 menunjukkan bahwa apabila sensor mendeteksi kendaraan maka sensor akan berwarna merah dan sebaliknya. Berikut gambaran jembatan dalam kondisi macet:



Gambar 4. 2 Jembatan dalam keadaan macet

Pada gambar 4.2 menunjukkan bahwa kondisi jembatan dalam kondisi macet. Jumlah kendaraan yang ada di jembatan banyak sehingga kendaraan satu dengan kendaraan yang lainnya tidak bisa saling mendahului. Pada gambar 4.2 terlihat bahwa jumlah kendaraan yang ada di jembatan sebanyak 8 kendaraan untuk lajur kiri dan 8 kendaraan untuk lajur kanan. K₁L, K₂L, K₇L, K₈L sebagai kendaraan terpanjang sedangkan K₃L, K₄L, K₅L, K₆L, K₇L, K₈L, K₁R, K₂R, K₃R, K₄R sebagai kendaraan dengan ukuran kecil serta K₅R dan K₆R sebagai kendaraan dengan ukuran sedang. Sensor yang

aktif 6 atau lebih bergantung posisi kendaraan saat berada di jembatan. Apabila kondisi jembatan macet maka jenis K_1L , K_2L , K_7L , K_8L hanya bisa 8 kendaraan dalam tiap jalur jembatan. Hal ini disebabkan panjang kendaraan terpanjang adalah 16,5 meter dan jarak aman adalah 5 meter. Hal ini yang menyebabkan jarak peletakan sensor akselerometer adalah 21,5 meter.

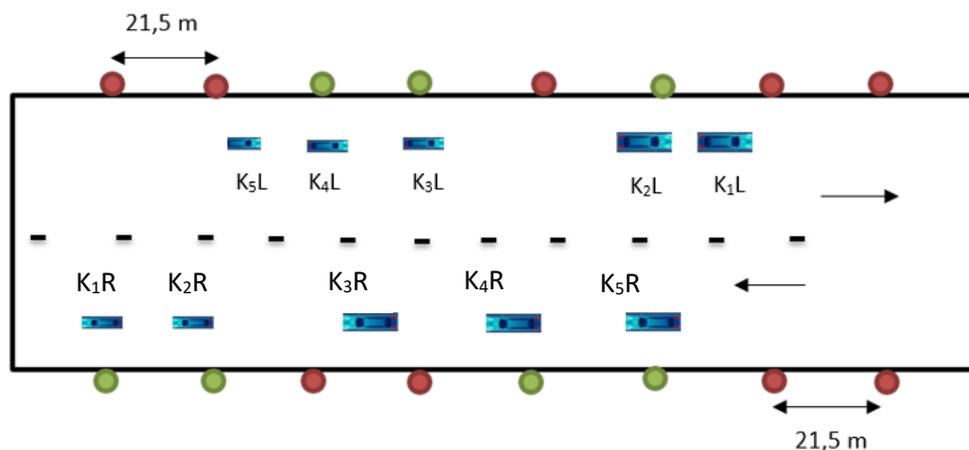
C. Algoritma untuk kondisi jembatan normal

Kendaraan yang melewati jembatan dalam kondisi legang memiliki kecepatan rata-rata 40 km/jam. Dalam kondisi ini, kendaraan tidak dapat mendahului kendaraan yang berada didepannya. Alur algoritma pada kondisi jembatan legang adalah sebagai berikut:

1. Kendaraan(n) masuk pada lajur kiri atau kanan jembatan
2. Sensor *loadcell* yang terletak disisi kanan dan kiri jembatan membaca berat kendaraan(n) yang masuk dari sisi kanan atau kiri jembatan.
3. Data beban kendaraan(n) yang masuk dari sisi kanan atau kiri jembatan tersimpan pada *buffer*.
4. Kendaraan(n) dari sisi kanan atau kiri jembatan terdeteksi oleh sensor akselerometer 1.
5. Beban total kendaraan (m) = Beban total kendaraan (m-1) + beban kendaraan yang masuk dari sisi kiri+ beban kendaraan kendaraan masuk dari sisi kanan.
6. Kendaraan (n+1) memasuki jembatan dari sisi kanan dan kiri, kembali ke langkah 2.
7. Kendaraan(n) dari sisi kanan atau kiri jembatan terdeteksi oleh sensor akselerometer 2.
8. Kendaraan (n+1) terdeteksi oleh sensor akselerometer 3.
9. Kendaraan (n+2) memasuki jembatan dari sisi kiri atau kanan jembatan, kembali ke langkah 2.
10. Kendaraan (n+n) memasuki jembatan dari sisi kiri atau kanan jembatan, kembali ke langkah 2.
11. Kendaraan (n+1), (n+2) dan (n+n) tidak dapat mendahului kendaraan (n).

12. Kendaraan (n), (n+1), (n+2) dan (n+n) memiliki kecepatan kurang lebih 40 km/jam karena kondisi jembatan normal, kembali ke langkah 1.
13. Apabila beban total kendaraan (m) melebihi kapasitas beban maksimal jembatan maka portal tertutup, menunggu hingga beban total kendaraan (m) tidak melebihi kapasitas beban maksimal.
14. Apabila beban total kendaraan (m) tidak melebihi kapasitas beban maksimal jembatan maka portal terbuka, kembali ke langkah 1.
15. Data beban total kendaraan (m) dikirim ke website sesuai dengan kondisi jembatan real time.

Pada simulasi visual basic terdapat *Human Machine Interface* (HMI) yang menunjukkan gambaran ilustrasi simulasi dari sistem monitoring dan kontrol jembatan untuk kondisi lengang. Sensor akselerometer yang berada di sepanjang jembatan difungsikan untuk mendeteksi posisi kendaraan. Dapat dilihat pada gambar 4.3 menunjukkan bahwa apabila sensor mendeteksi kendaraan maka sensor akan berwarna merah dan sebaliknya. Berikut gambaran jembatan dalam kondisi normal:



Gambar 4. 3 Jembatan dalam kondisi normal

Pada gambar 4.3 menunjukkan bahwa kondisi jembatan dalam kondisi normal. Jumlah kendaraan yang ada di jembatan banyak sehingga kendaraan satu dengan kendaraan yang lainnya tidak bisa saling mendahului. Pada gambar 4.3 terlihat bahwa jumlah kendaraan yang ada di jembatan sebanyak 5 kendaraan untuk lajur kiri dan 8 kendaraan untuk lajur kanan. K₁L, K₂L, K₄R, K₅R sebagai kendaraan terpanjang sedangkan K₁R, K₂R, K₃L, K₄L, K₅L

sebagai kendaraan dengan ukuran kecil serta K₃R dan K₄R sebagai kendaraan dengan ukuran sedang. Sensor yang aktif 4 atau lebih bergantung posisi kendaraan saat berada di jembatan. Apabila kondisi jembatan macet maka jenis K₁L, K₂L, K₄R, K₅R hanya bisa 8 kendaraan dalam tiap jalur jembatan. Hal ini disebabkan panjang kendaraan terpanjang adalah 16,5 meter dan jarak aman adalah 5 meter. Hal ini yang menyebabkan jarak peletakan sensor akselerometer adalah 21,5 meter.

4.1.2 Fungsi-fungsi pada alur algoritma

Pada subbab alur algoritma telah dijelaskan mengenai urutan langkah-langkah program untuk masing-masing kondisi jembatan. Pada subbab ini menjelaskan beberapa penjelasan mengenai fungsi-fungsi pada alur algoritma sebagai berikut:

A. Fungsi penimbangan beban kendaraan

Data kendaraan yang ada didalam *excell* diasumsikan sebagai data yang dapat mewakili data kendaraan real. Data yang dipersiapkan ada 30 data kendaraan yang terdiri dari 3 jenis data kendaraan yaitu data truk, data kendaraan pribadi dan data bus. Ketika menekan *button loadcell* pada *Human machine interface* (HMI) pada *visual basic* maka data tersebut akan diproses sesuai dengan kondisi jembatan yang dipilih. Sehingga langkah program yang pertama adalah kendaraan memasuki jembatan dan proses penimbangan kendaraan.

B. Fungsi pendeteksian posisi kendaraan

Fungsi pendeteksian posisi kendaraan dilakukan setelah program melakukan langkah pertama yaitu proses penimbangan berat kendaraan. Pada fungsi ini, kendaraan yang memasuki jembatan akan melewati 8 sensor akselerometer. Ketika kendaraan pertama memasuki jembatan maka sensor akselerometer pertama akan aktif. Sensor akselerometer dapat dikatakan aktif apabila menampilkan warna hijau dan sebaliknya. Sedangkan untuk kendaraan yang kedua akan melakukan proses penimbangan terlebih dahulu sebelum melewati sensor akselerometer pertama.

C. Fungsi penghapusan data kendaraan

Fungsi penghapusan data kendaraan ini dilakukan bersamaan dengan proses pendeteksian posisi kendaraan. Ketika kendaraan pertama sudah melewati sensor akselerometer 8, maka data yang tersimpan akan terhapus.

D. Fungsi penutupan portal jembatan

Proses penutupan portal jembatan dilakukan setelah proses penjumlahan data kendaraan dilakukan. Apabila data yang tersimpan melebihi kapasitas beban maksimal maka portal akan tertutup. Portal pada *Human Machine Interface* (HMI) ditandai dengan 2 warna yaitu merah dan hijau. Apabila total data berat kendaraan melebihi kapasitas beban maksimal jembatan maka led portal akan berwarna merah.

E. Fungsi pembukaan portal jembatan

Proses penutupan portal jembatan dilakukan setelah proses penjumlahan data kendaraan dilakukan. Apabila proses penjumlahan data kendaraan tidak melebihi kapasitas beban maksimal maka portal akan terbuka dan led portal pada *Human Machine Interface* (HMI) akan berwarna hijau.

F. Fungsi acak data kendaraan

Fungsi acak data kendaraan ini dapat diartikan bahwa data yang diproses oleh sistem akan diambil secara acak atau random. Hal ini dilakukan karena kendaraan yang masuk jembatan tidak dapat diprediksi. Data kendaraan yang ada didalam *excell* diasumsikan sebagai data yang dapat mewakili data kendaraan real. Data yang dipersiapkan ada 30 data kendaraan yang terdiri dari 3 jenis data kendaraan yaitu data truk, data kendaraan pribadi dan data bus.

4.1.3 Pengujian algoritma menggunakan perangkat lunak *visual basic*

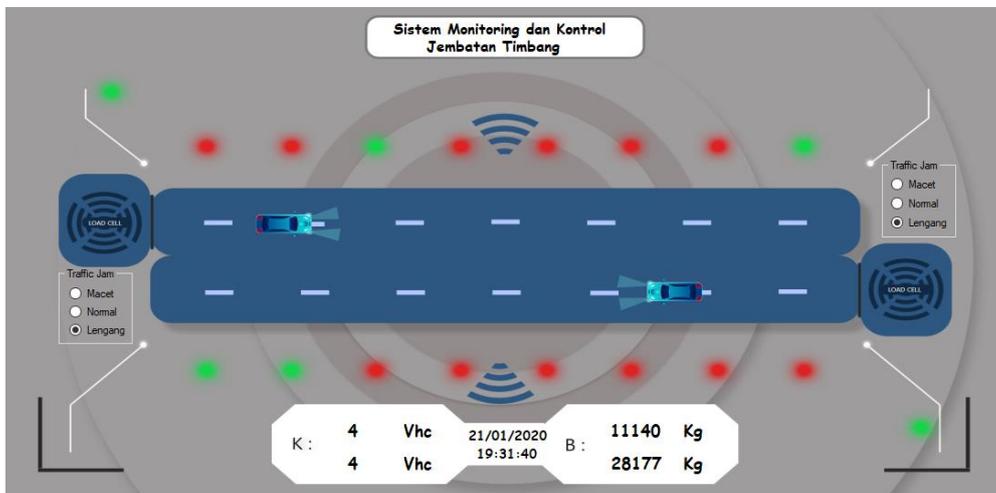
Alur algoritma program dan penjelasan fungsi-fungsi pada algoritma program sudah dijelaskan, setelah itu dilakukan pengujian algoritma menggunakan perangkat lunak *visual basic* untuk membuktikan kebenaran terhadap algoritma yang sudah ditetapkan. Berdasarkan kondisi jembatan maka dilakukan beberapa pengujian algoritma diantaranya:

A. Pengujian algoritma terhadap jembatan lengang

Pada kondisi lengang memungkinkan kendaraan memiliki kecepatan tinggi yaitu 60 km/jam, sehingga kendaraan yang satu dan lainnya dapat saling mendahului.

- a. Pengujian algoritma terhadap jembatan lengang dengan beban kendaraan tidak melebihi kapasitas maksimal jembatan.

Berikut merupakan hasil salah satu pengujian algoritma pada kondisi jembatan lengang dengan beban kendaraan tidak melebihi kapasitas maksimal jembatan:



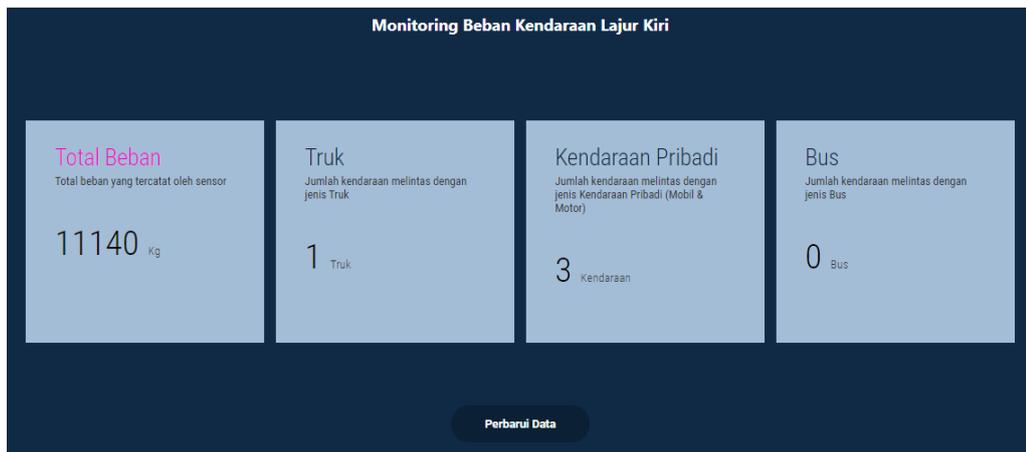
Gambar 4. 4 Jembatan dalam kondisi lengang

Pada gambar 4.4 menunjukkan bahwa jembatan dalam kondisi lengang, dapat ditandai dengan 4 kendaraan pada masing-masing lajur jembatan. Untuk lajur kanan terdapat 4 kendaraan dan lajur kiri terdapat 4 kendaraan. Total beban kendaraan yang berada di jembatan untuk kedua lajur adalah 39.317 kg. Dalam kondisi ini, beban kendaraan yang berada diatas jembatan tidak melebihi kapasitas maksimal jembatan sehingga portal pada ujung jembatan berwarna hijau yang artinya posisi portal sedang terbuka dan kendaraan yang lainnya bisa memasuki jembatan.



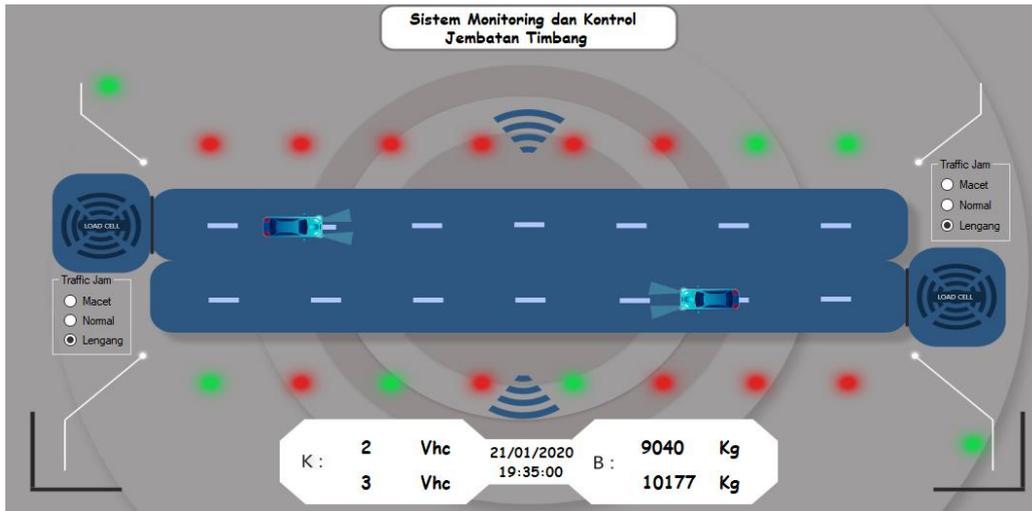
Gambar 4. 5 Tampilan informasi kendaraan pada lajur kanan

Pada gambar 4.5 menunjukkan bahwa terdapat 2 jenis kendaraan yaitu truk dan kendaraan pribadi dengan total beban keempat kendaraan tersebut adalah 28.177 kg.



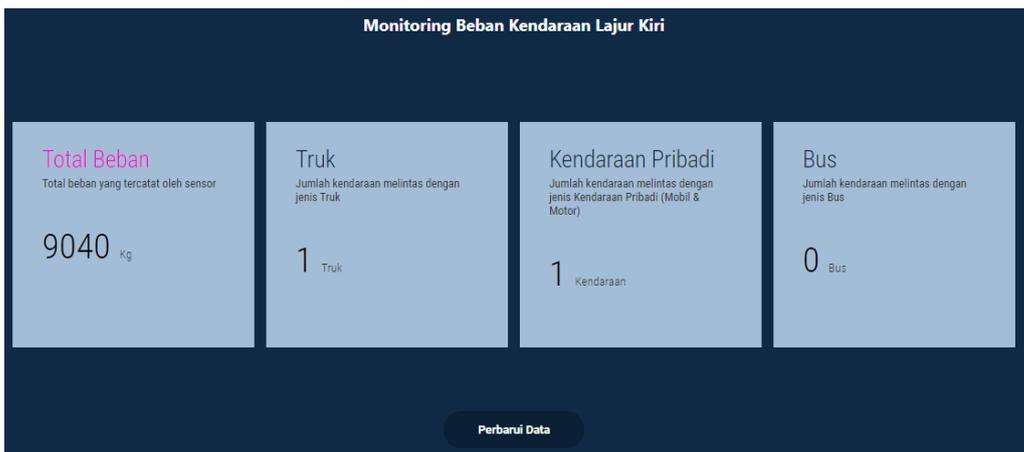
Gambar 4. 6 Tampilan informasi kendaraan pada lajur kanan

Pada gambar 4.6 menunjukkan bahwa terdapat 2 jenis kendaraan yaitu truk dan kendaraan pribadi. Jumlah truk adalah 1 dan jumlah kendaraan pribadi adalah 3. Total beban kendaraan untuk lajur kiri adalah 11.140 kg. Ketika kendaraan pada lajur kanan dan lajur kiri keluar jembatan, maka beban kendaraan pada kedua jembatan juga berkurang. Untuk lebih jelasnya dapat dilihat gambar 4.7.



Gambar 4. 7 Jembatan dalam kondisi lengang

Pada gambar 4.7 menunjukkan bahwa jembatan jembatan masih dalam kondisi lengang dengan beban kapasitas kendaraan tidak melebihi kapasitas maksimal jembatan.



Gambar 4. 8 Tampilan informasi kendaraan pada lajur kiri

Pada gambar 4.8 menunjukkan bahwa terdapat 2 jenis kendaraan yaitu truk dan kendaraan pribadi dengan total beban kedua kendaraan tersebut adalah 9.040 kg. Kendaraan yang keluar jembatan adalah 1 truk dan 1 mobil, hal ini dapat dibuktikan dengan melihat gambar 4.9.

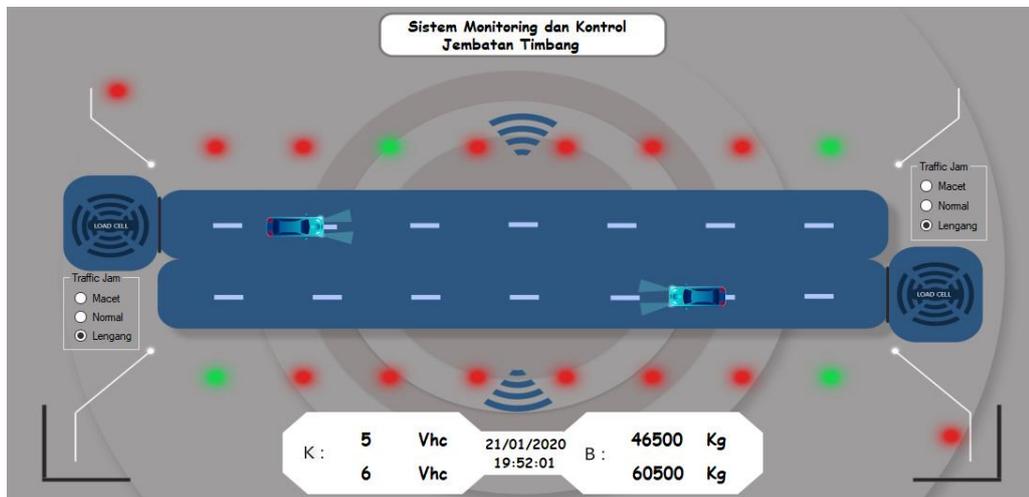


Gambar 4.9 Tampilan informasi kendaraan pada lajur kanan

Pada gambar 4.9 menunjukkan bahwa terdapat 2 jenis kendaraan yaitu truk dan kendaraan pribadi. Jumlah truk adalah 1 dan jumlah kendaraan pribadi adalah 2. Total beban kendaraan untuk lajur kiri adalah 10.177 kg.

- b. Pengujian algoritma terhadap jembatan lengang dengan beban kendaraan melebihi kapasitas maksimal jembatan.

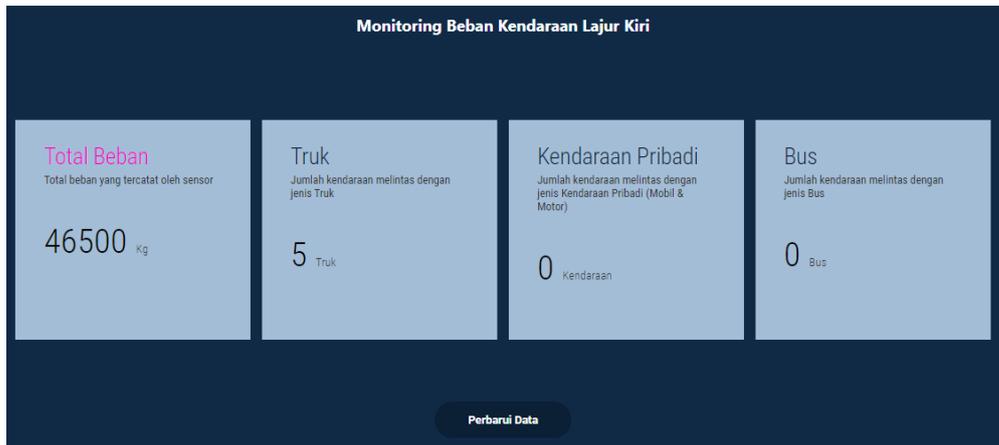
Berikut merupakan hasil salah satu pengujian algoritma pada kondisi jembatan lengang dengan beban kendaraan melebihi kapasitas maksimal jembatan:



Gambar 4.10 Pengujian algoritma terhadap kondisi jembatan lengang

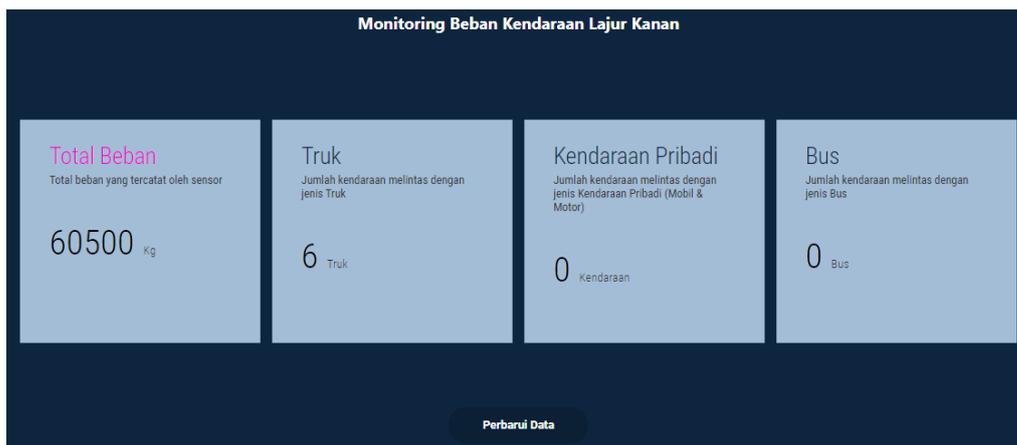
Pada gambar 4.10 menunjukkan bahwa jembatan dalam kondisi lengang namun beban kendaraan yang berada di jembatan melebihi kapasitas maksimal jembatan. Pada lajur kiri terdapat 5 kendaraan dengan jenis yang berbeda-beda. Sedangkan pada lajur kanan juga terdapat 6 kendaraan yang berbeda-beda.

Sehingga beban kendaraan yang berada di jembatan melebihi kapasitas maksimal jembatan yaitu 107.000 kg. Hal ini dikarenakan kapasitas maksimal jembatan yang ditetapkan adalah 100.000 kg untuk kedua sisi jembatan. Portal pada sisi kedua jembatan tertutup karena beban kendaraan melebihi kapasitas jembatan yang telah ditetapkan.



Gambar 4. 11 Tampilan informasi kendaraan pada lajur kiri

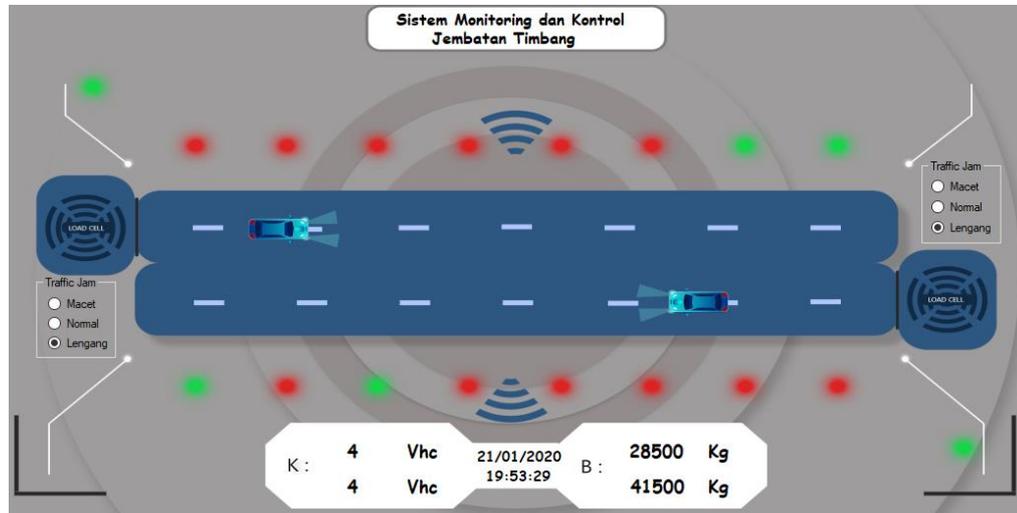
Pada gambar 4.11 menunjukkan bahwa pada lajur kiri terdapat 1 jenis kendaraan yaitu 5 jenis truk yang berbeda-beda. Dikarenakan total beban dari 5 truk tersebut melebihi kapasitas beban jembatan maka portal pada lajur kiri tertutup. Hal ini menyebabkan kendaraan lain tidak bisa memasuki jembatan untuk beberapa saat hingga beban kendaraan tidak melebihi kapasitas beban jembatan.



Gambar 4. 12 Tampilan informasi kendaraan pada lajur kanan

Pada gambar 4.12 menunjukkan bahwa pada lajur kanan terdapat 1 jenis kendaraan yaitu 6 jenis truk yang berbeda-beda. Dikarenakan total beban dari 6 truk tersebut melebihi kapasitas beban jembatan maka portal pada lajur kanan

tertutup. Hal ini menyebabkan kendaraan lain tidak bisa memasuki jembatan untuk beberapa saat hingga beban kendaraan tidak melebihi kapasitas beban jembatan. Namun ketika kendaraan beberapa truk keluar dari jembatan, beban truk yang berada diatas jembatan berkurang.



Gambar 4. 13 Pengujian algoritma terhadap kondisi jembatan lengang

Pada gambar 4.13 menunjukkan bahwa kendaraan yang berada diatas jembatan tidak melebihi kapasitas beban jembatan. Hal ini dikarenakan beberapa truk sudah keluar dari jembatan. Untuk lajur kiri ada 1 truk yang keluar sedangkan untuk lajur kanan ada 2 truk yang keluar jembatan. Sehingga total beban truk yang berada dijembatan adalah 70.000kg.



Gambar 4. 14 Tampilan informasi kendaraan pada lajur kiri

Pada gambar 4.14 menunjukkan bahwa pada lajur kiri terdapat 1 jenis kendaraan yaitu 4 jenis truk yang berbeda-beda. Dikarenakan total beban dari 4 truk tersebut tidak melebihi kapasitas beban jembatan dengan beban 28.500 kg

maka portal pada lajur kiri terbuka sehingga kendaraan lain bisa memasuki jembatan.



Gambar 4. 15 Tampilan informasi kendaraan pada lajur kanan

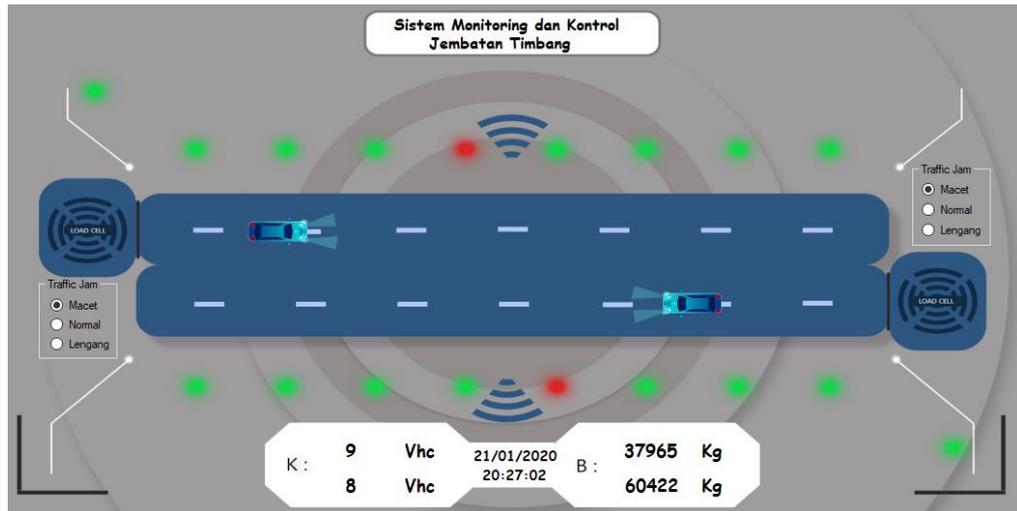
Pada gambar 4.15 menunjukkan bahwa pada lajur kiri terdapat 1 jenis kendaraan yaitu 4 jenis truk yang berbeda-beda. Dikarenakan total beban dari 4 truk tersebut tidak melebihi kapasitas beban jembatan dengan beban 41.500 kg maka portal pada lajur kanan terbuka sehingga kendaraan lain bisa memasuki jembatan.

B. Pengujian algoritma terhadap jembatan macet

Pada kondisi macet memungkinkan kendaraan memiliki kecepatan rendah yaitu 5 km/jam, sehingga kendaraan yang satu dan lainnya tidak dapat saling mendahului.

- a. Pengujian algoritma terhadap jembatan macet dengan beban kendaraan tidak melebihi kapasitas maksimal jembatan.

Berikut merupakan hasil salah satu pengujian algoritma pada kondisi jembatan lengang dengan beban kendaraan tidak melebihi kapasitas maksimal jembatan:



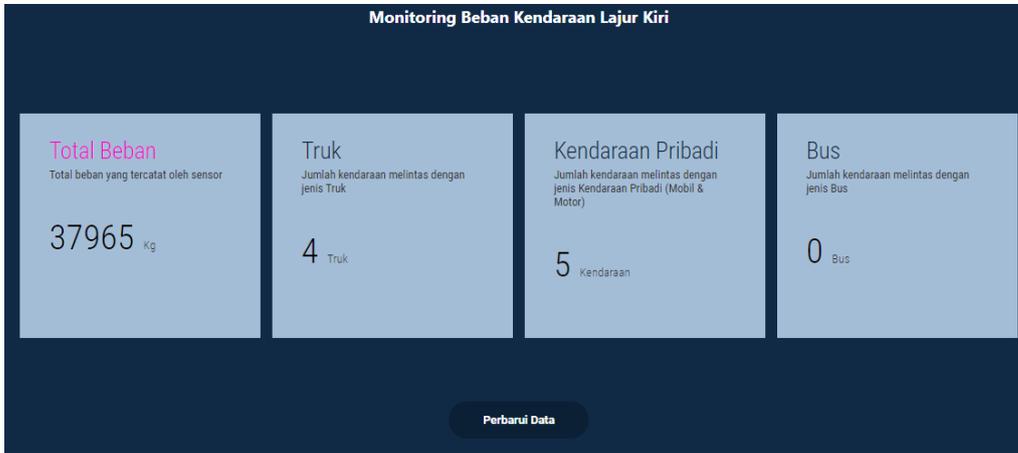
Gambar 4. 16 Pengujian algoritma terhadap kondisi jembatan macet

Pada gambar 4.16 menunjukkan bahwa jembatan dalam kondisi macet namun beban kendaraan yang berada di jembatan tidak melebihi kapasitas maksimal jembatan. Pada lajur kiri terdapat 9 kendaraan dan 8 kendaraan pada lajur kanan dengan jenis yang berbeda-beda. Total beban kendaraan yang berada di jembatan tidak melebihi kapasitas maksimal jembatan yaitu 98.387 kg. Hal ini dikarenakan kapasitas maksimal jembatan yang ditetapkan adalah 100.000 kg untuk kedua sisi jembatan. Portal pada sisi kedua jembatan tetap terbuka karena beban kendaraan tidak melebihi kapasitas jembatan yang telah ditetapkan sehingga kendaraan lain bisa memasuki jembatan. Untuk mengetahui jenis kendaraan yang melewati jembatan pada lajur kanan maka dapat dilihat pada gambar 4.17.



Gambar 4. 17 Tampilan informasi kendaraan pada lajur kanan

Pada gambar 4.17 menunjukkan bahwa terdapat 3 jenis kendaraan yang berada di jembatan diantaranya 2 jenis truk, 5 jenis kendaraan pribadi dan 1 jenis bus. Total beban kendaraan pada lajur kanan adalah 60.422 kg.

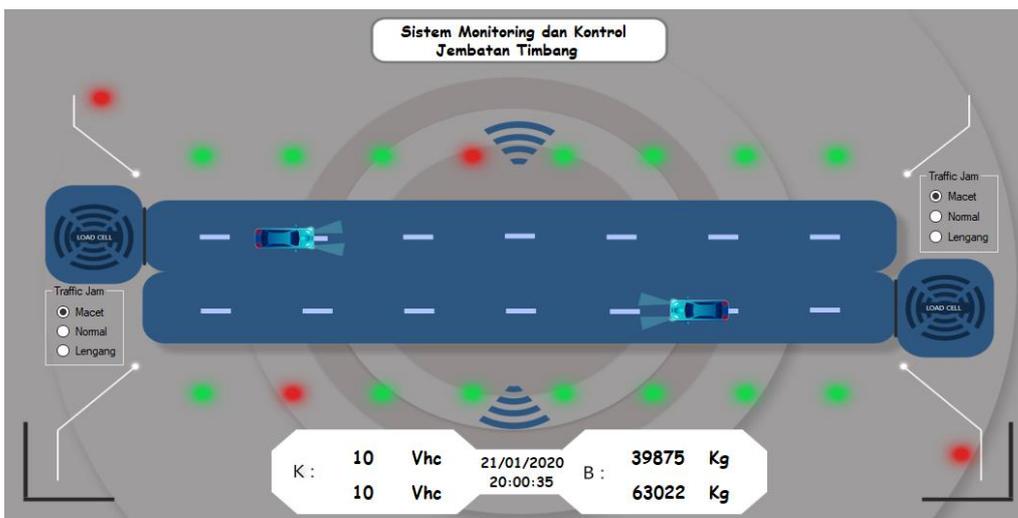


Gambar 4. 18 Tampilan informasi kendaraan pada lajur kiri

Pada gambar 4.18 menunjukkan bahwa terdapat 2 jenis kendaraan yang berada di jembatan diantaranya 4 jenis truk dan 5 jenis kendaraan. Total beban kendaraan pada lajur kanan adalah 37.965 kg.

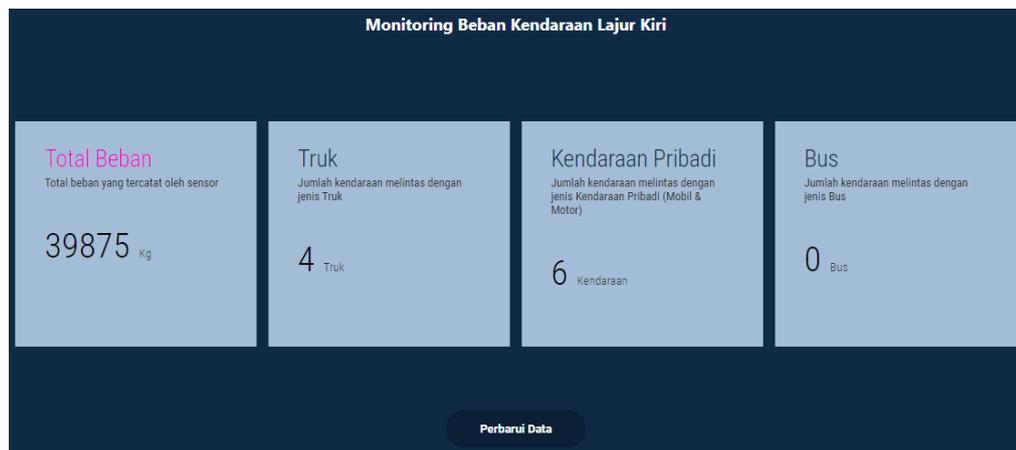
b. Pengujian algoritma terhadap jembatan macet dengan beban kendaraan melebihi kapasitas maksimal jembatan.

Berikut merupakan hasil salah satu pengujian algoritma pada kondisi jembatan macet dengan beban kendaraan melebihi kapasitas maksimal jembatan:



Gambar 4. 19 Pengujian algoritma terhadap kondisi jembatan macet

Pada gambar 4.19 menunjukkan bahwa jembatan dalam kondisi macet namun beban kendaraan yang berada di jembatan melebihi kapasitas maksimal jembatan. Pada lajur kiri terdapat 10 kendaraan dan 10 kendaraan pada lajur kanan dengan jenis yang berbeda-beda. Total beban kendaraan yang berada di jembatan tidak melebihi kapasitas maksimal jembatan yaitu 98.387 kg. Hal ini dikarenakan kapasitas maksimal jembatan yang ditetapkan adalah 102.897 kg untuk kedua sisi jembatan. Beban kendaraan melebihi kapasitas beban maksimal jembatan yang telah ditetapkan yaitu 100.000kg. Portal pada sisi kedua jembatan terturup karena beban kendaraan melebihi kapasitas jembatan yang telah ditetapkan sehingga kendaraan lain tidak bisa memasuki jembatan. Untuk mengetahui jenis kendaraan yang melewati jembatan pada lajur kiri maka dapat dilihat pada gambar 4.20.



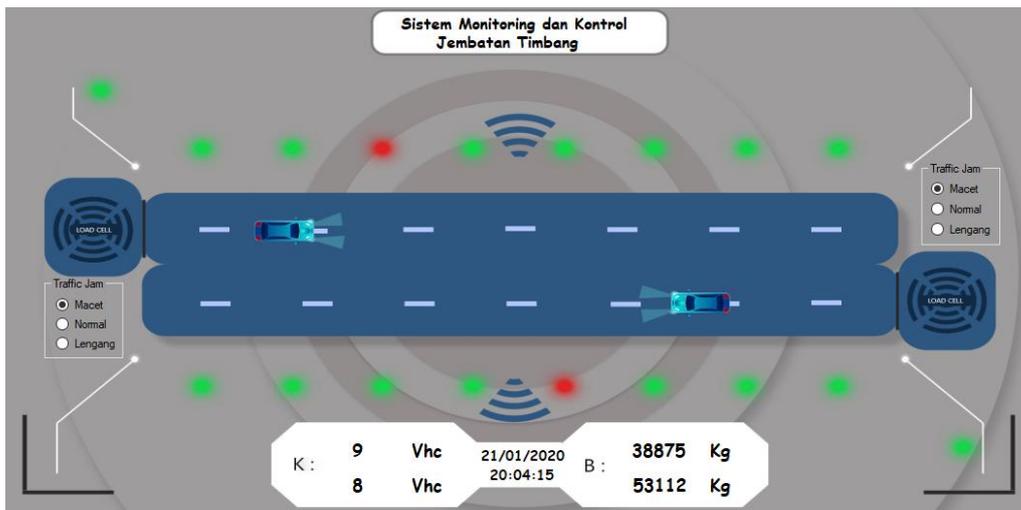
Gambar 4. 20 Tampilan informasi kendaraan pada lajur kiri

Pada gambar 4.20 menunjukkan bahwa terdapat 2 jenis kendaraan yang berada di jembatan diantaranya 4 jenis truk dan 6 jenis kendaraan pribadi. Total beban kendaraan pada lajur kiri adalah 39.875 kg. Untuk mengetahui informasi jenis kendaraan yang melewati jembatan lajur kanan maka dapat dilihat gambar 4.21.



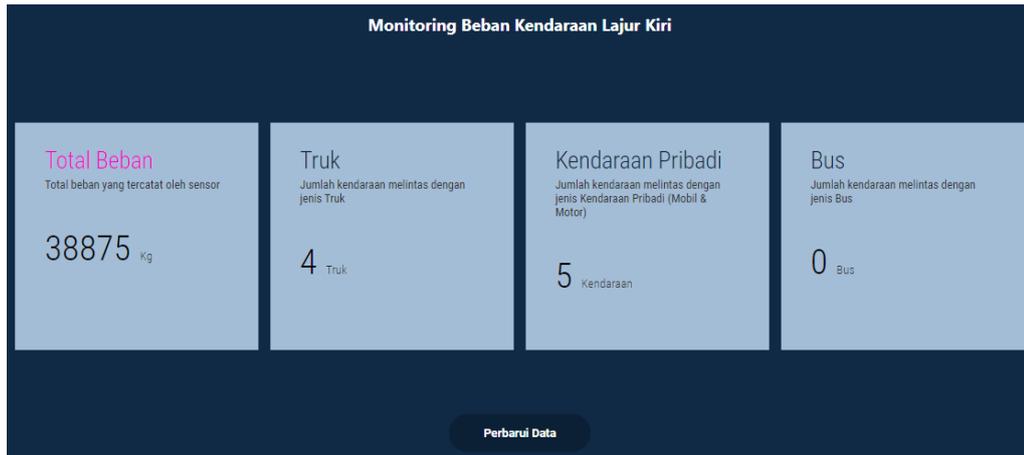
Gambar 4. 21 Tampilan informasi kendaraan pada lajur kanan

Pada gambar 4.21 menunjukkan bahwa terdapat 3 jenis kendaraan yang berada di jembatan diantaranya 2 jenis truk dan 7 jenis kendaraan pribadi dan 1 bus. Total beban kendaraan pada lajur kiri adalah 63.022 kg. Namun beberapa kendaraan sudah keluar dari jembatan, dapat dilihat pada gambar 4.19.



Gambar 4. 22 Pengujian algoritma terhadap kondisi jembatan macet

Pada gambar 4.22 menunjukkan bahwa jembatan tidak mengalami overload karena beban kendaraan yang memasuki jembatan tidak melebihi kapasitas jembatan yaitu dari lajur kiri 38.875 kg dan dari lajur kanan 53.112 kg. Total beban seluruh kendaraan adalah 91.987 kg. Untuk mengetahui jenis kendaraan yang melewati jembatan pada lajur kiri maka dapat dilihat pada gambar 4.23.



Gambar 4. 23 Tampilan informasi kendaraan pada lajur kiri

Pada gambar 4.23 menunjukkan bahwa terdapat 3 jenis kendaraan yang berada di jembatan diantaranya 4 jenis truk dan 6 jenis kendaraan pribadi. Total beban kendaraan pada lajur kiri adalah 38.875 kg. Untuk mengetahui informasi jenis kendaraan yang melewati jembatan lajur kanan maka dapat dilihat gambar 4.24.



Gambar 4. 24 Tampilan informasi kendaraan pada lajur kanan

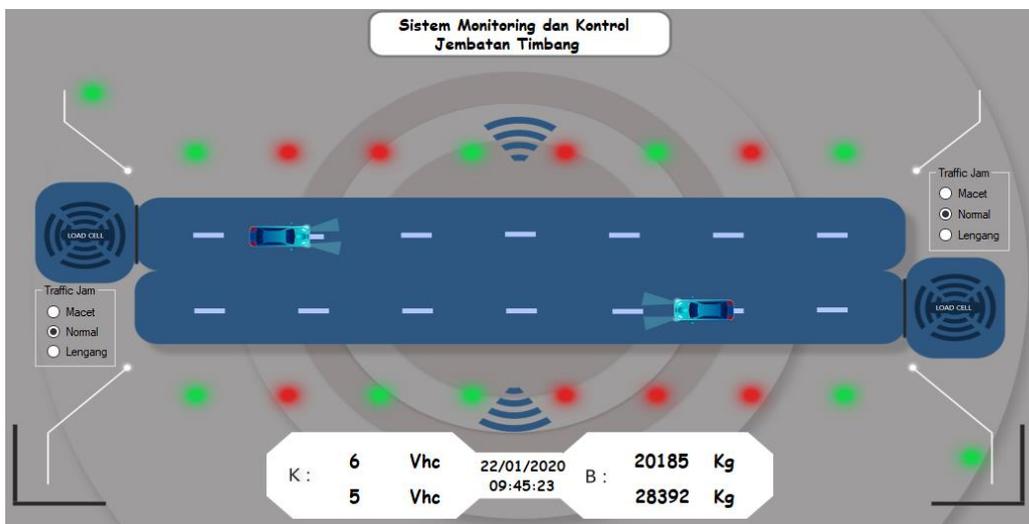
Pada gambar 4.24 menunjukkan bahwa terdapat 3 jenis kendaraan yang berada di jembatan diantaranya 1 jenis truk dan 6 jenis kendaraan pribadi dan 1 bus. Total beban kendaraan pada lajur kiri adalah 53.112 kg. Total beban kendaraan dari kedua lajur adalah 91.987 kg yang tidak melebihi kapasitas maksimal jembatan.

C. Pengujian algoritma terhadap jembatan kondisi normal

Pada kondisi jembatan normal memungkinkan kendaraan memiliki kecepatan sedang yaitu 40 km/jam, sehingga kendaraan yang satu dan lainnya tidak dapat saling mendahului.

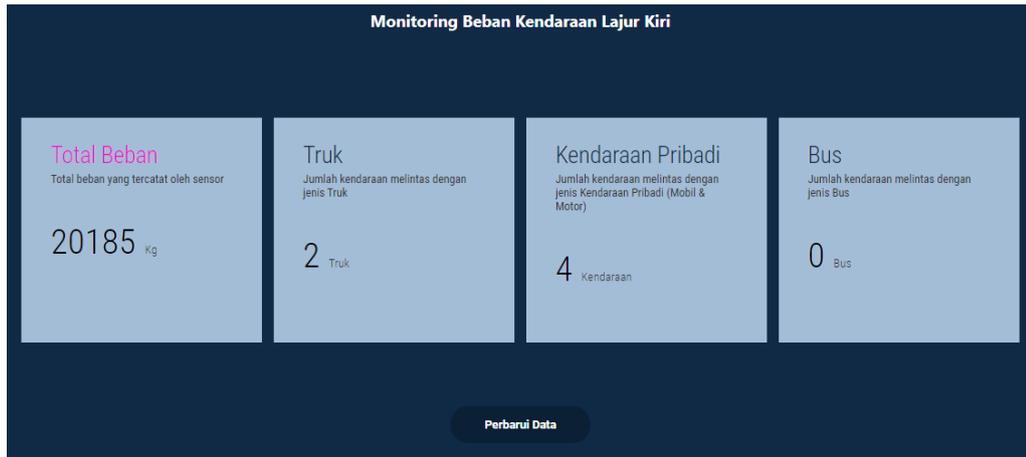
- a. Pengujian algoritma terhadap jembatan kondisi normal dengan beban kendaraan tidak melebihi kapasitas maksimal jembatan.

Berikut merupakan hasil salah satu pengujian algoritma pada kondisi jembatan normal dengan beban kendaraan tidak melebihi kapasitas maksimal jembatan:



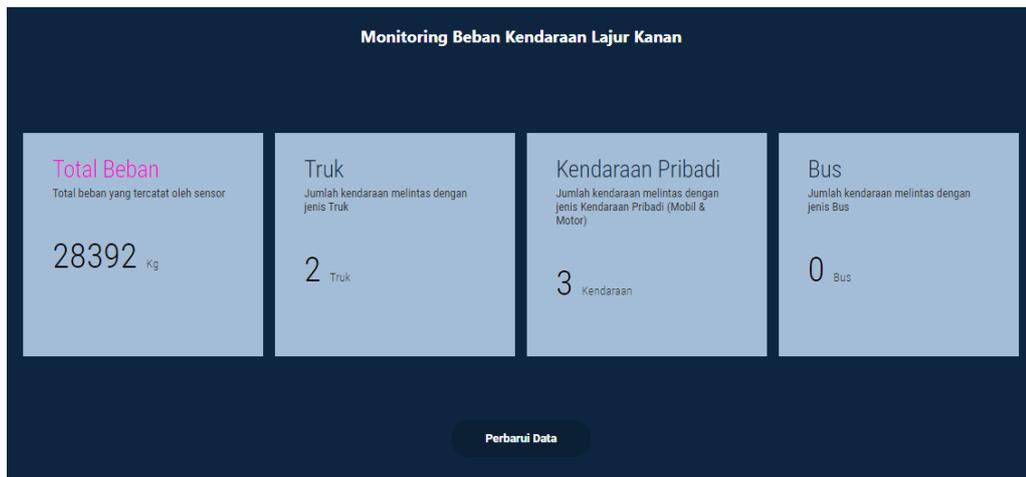
Gambar 4. 25 Pengujian algoritma terhadap kondisi jembatan normal

Pada gambar 4.25 menunjukkan bahwa jembatan dalam kondisi normal dengan 11 kendaraan untuk kedua lajur. Untuk lajur kanan terdapat 6 kendaraan dan lajur kiri terdapat 5 kendaraan. Total beban kendaraan yang berada di jembatan untuk kedua lajur adalah 48.577 kg. Pada kondisi ini beban kendaraan yang berada diatas jembatan tidak melebihi kapasitas maksimal jembatan sehingga portal pada ujung jembatan berwarna hijau yang artinya posisi portal sedang terbuka. Untuk mengetahui jenis kendaraan yang melewati jembatan pada lajur kiri maka dapat dilihat pada gambar 4.26.



Gambar 4. 26 Tampilan informasi kendaraan pada lajur kiri

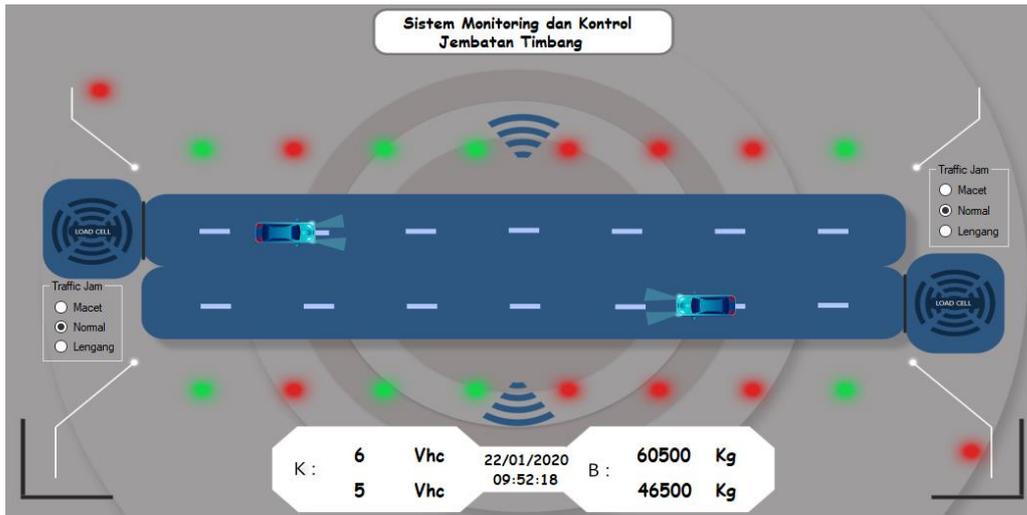
Pada gambar 4.26 menunjukkan bahwa terdapat 2 jenis kendaraan yang berada di jembatan diantaranya 2 jenis truk dan 4 jenis kendaraan pribadi. Total beban kendaraan pada lajur kiri adalah 20.185 kg. Untuk mengetahui informasi jenis kendaraan yang melewati jembatan lajur kanan maka dapat dilihat gambar 4.27.



Gambar 4. 27 Tampilan informasi kendaraan pada lajur kiri

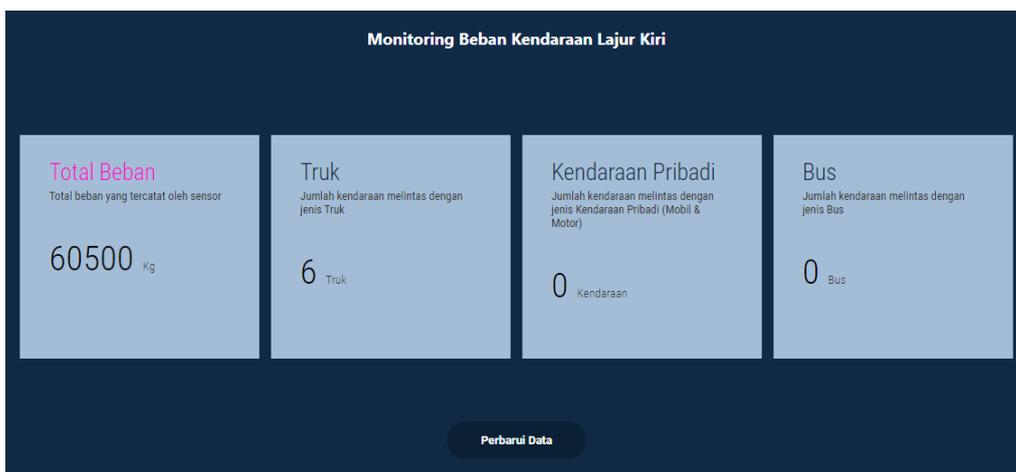
- b. Pengujian algoritma terhadap jembatan kondisi normal dengan beban kendaraan melebihi kapasitas maksimal jembatan.

Berikut merupakan hasil salah satu pengujian algoritma pada kondisi jembatan normal dengan beban kendaraan tidak melebihi kapasitas maksimal jembatan:



Gambar 4. 28 Pengujian algoritma terhadap kondisi jembatan normal

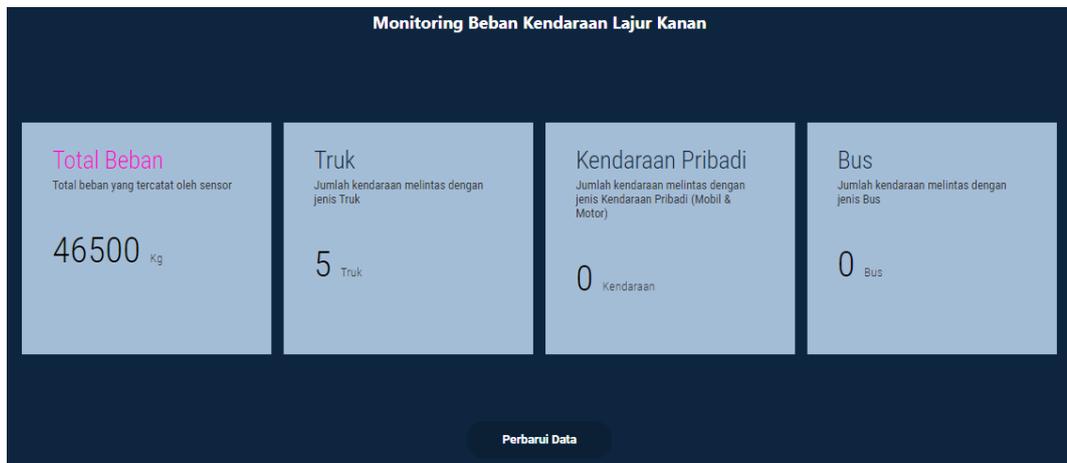
Pada gambar 4.28 menunjukkan bahwa jembatan dalam kondisi normal dengan 11 kendaraan untuk kedua lajur. Untuk lajur kanan terdapat 6 kendaraan dan lajur kiri terdapat 5 kendaraan. Total beban kendaraan yang berada di jembatan untuk kedua lajur adalah 107.000 kg. Pada kondisi ini beban kendaraan yang berada diatas jembatan melebihi kapasitas maksimal jembatan sehingga portal pada ujung jembatan berwarna hijau yang artinya posisi portal tertutup. Untuk mengetahui jenis kendaraan yang melewati jembatan pada lajur kiri maka dapat dilihat pada gambar 4.29 sebagai berikut:



Gambar 4. 29 Tampilan informasi kendaraan pada lajur kiri

Pada gambar 4.29 menunjukkan bahwa hanya ada 1 jenis kendaraan yaitu jenis truk. Total beban total truk pada lajur kiri adalah 60.500 kg. Untuk

mengetahui informasi jenis kendaraan yang melewati jembatan lajur kanan maka dapat dilihat gambar 4.30.



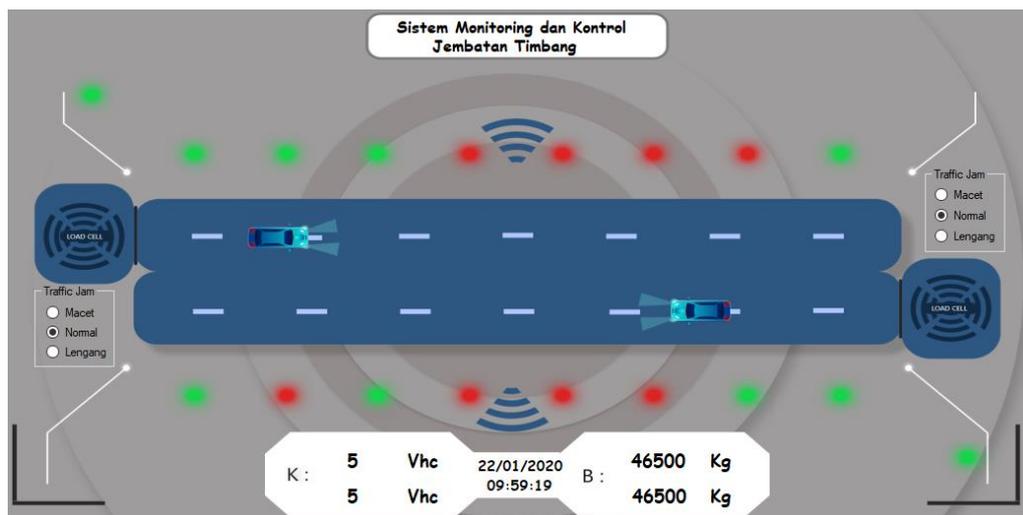
Gambar 4. 30 Tampilan informasi kendaraan pada lajur kanan

Pada gambar 4.30 menunjukkan bahwa hanya ada 1 jenis kendaraan yaitu jenis truk. Total beban total truk pada lajur kiri adalah 46.500 kg.

D. Pengujian algoritma terhadap terhadap kendaraan sejenis

a. Pengujian algoritma untuk kendaraan jenis truk

Berdasarkan muatannya truk dibedakan menjadi beberapa jenis yaitu truk berukuran besar, sedang dan kecil. Maka dilakukan pengujian algoritma terhadap data truk.



Gambar 4. 31 Pengujian algoritma dengan data truk

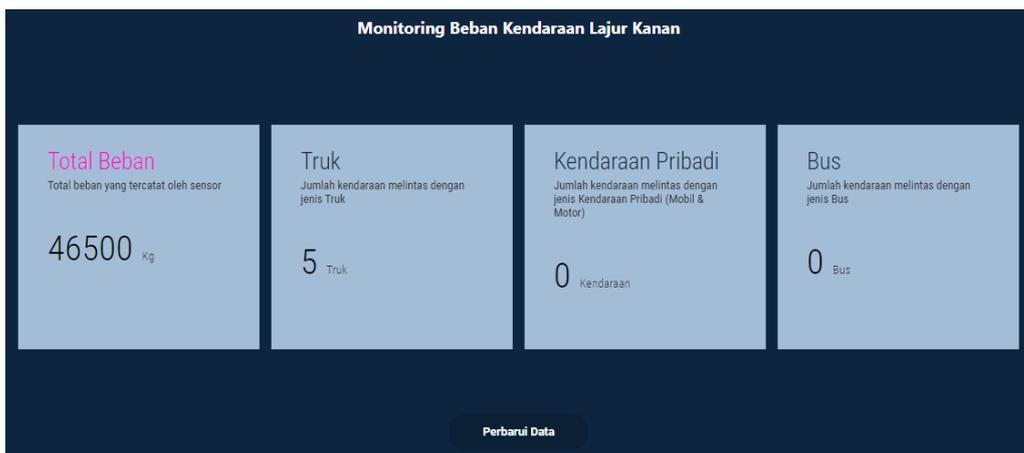
Pada gambar 4.31 menunjukkan bahwa terdapat 5 buah truk pada lajur kanan dan 5 buah truk pada lajur kiri. Total beban truk adalah 93.000 kg. Sehingga

beban kedua jalur jembatan tidak melebihi kapasitas maksimal jembatan. Jembatan dalam keadaan terbuka dengan ditandai warna hijau pada led portal.



Gambar 4. 32 Tampilan informasi kendaraan pada lajur kiri

Pada gambar 4.32 menunjukkan bahwa hanya ada 1 jenis kendaraan yaitu jenis truk. Total beban total truk pada lajur kiri adalah 46.500 kg.

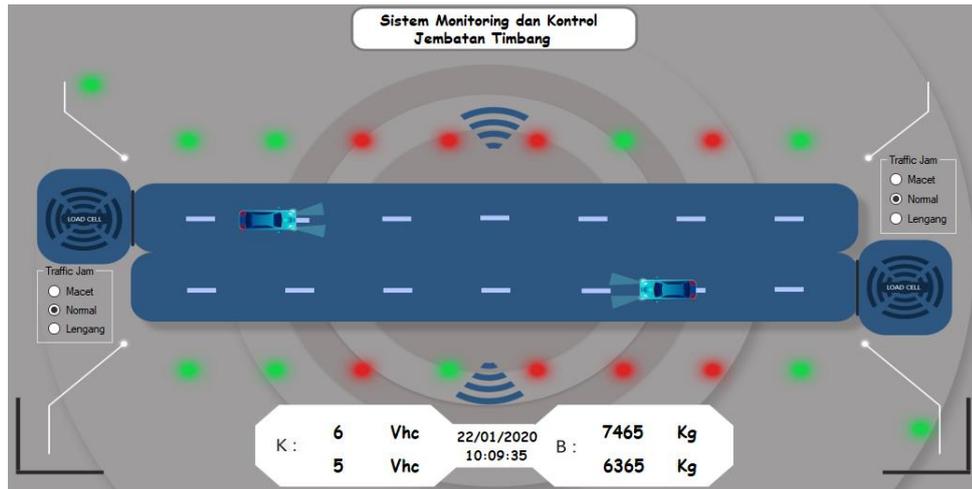


Gambar 4. 33 Tampilan informasi kendaraan pada lajur kanan

Pada gambar 4.33 menunjukkan bahwa hanya ada 1 jenis kendaraan yaitu jenis truk. Total beban total truk pada lajur kiri adalah 46.500 kg.

b. Pengujian algoritma untuk kendaraan pribadi

Kendaraan pribadi dibedakan beberapa jenis diantaranya yaitu mobil dan sepeda motor. Dilakukan pengujian algoritma terhadap kendaraan pribadi adalah sebagai berikut:



Gambar 4. 34 Pengujian algoritma dengan data kendaraan pribadi

Pada gambar 4.34 menunjukkan bahwa terdapat 6 kendaraan pribadi pada lajur kiri dan 6 lajur kanan dengan total 13.830kg sehingga beban kedua jalur tidak melebihi kapasitas maksimal jembatan. Informasi kendaraan pada jalur kanan dapat dilihat pada gambar 4.35.



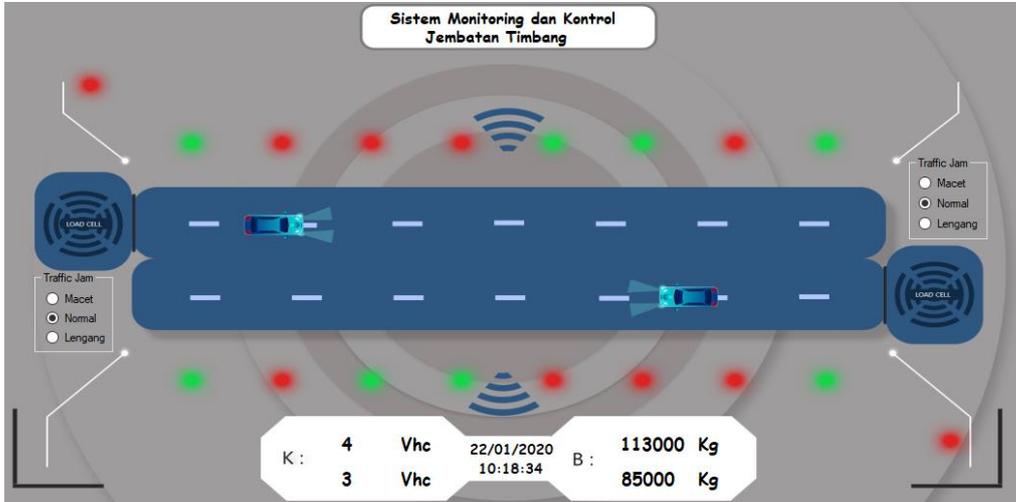
Gambar 4. 35 Tampilan informasi kendaraan pada lajur kiri



Gambar 4. 36 Tampilan informasi kendaraan pada lajur kanan

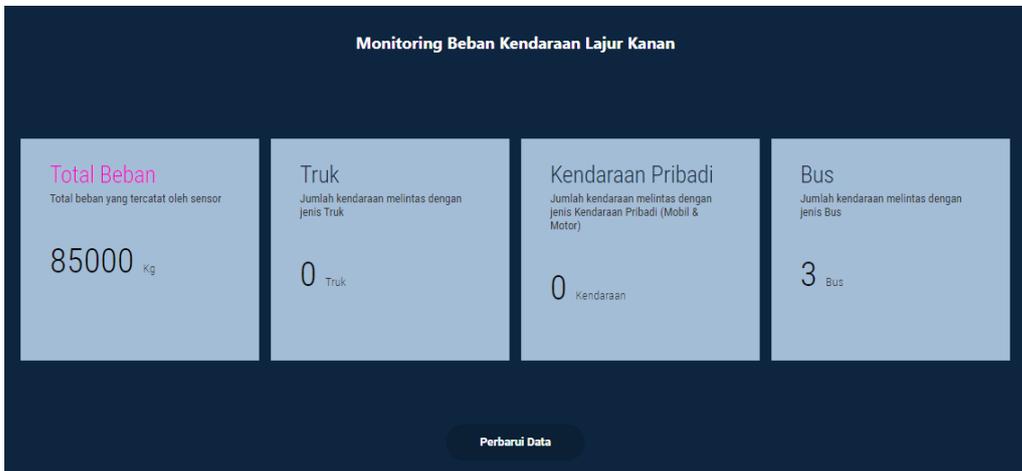
c. Pengujian algoritma untuk bus

Pengujian algoritma untuk kendaraan bus juga dilakukan. Dapat dilihat pada gambar 3.37 sebagai berikut:



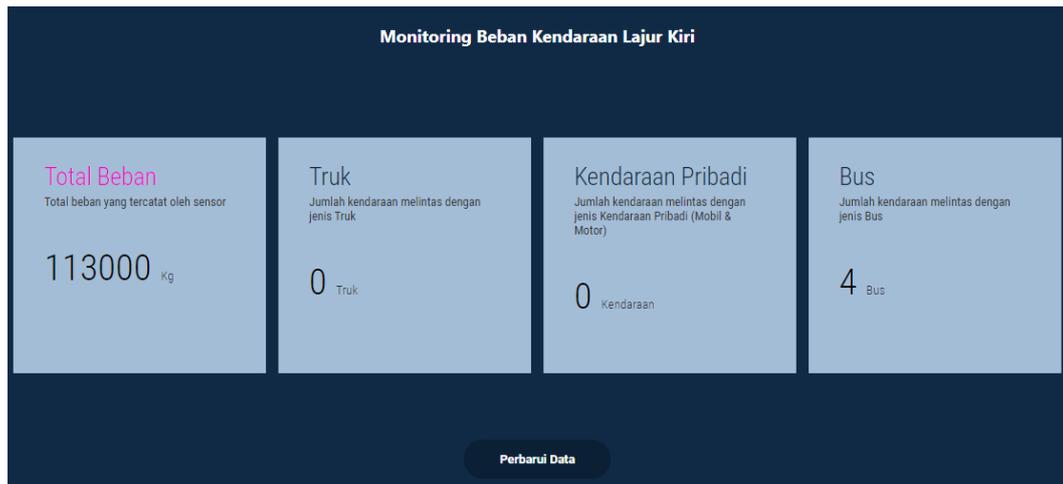
Gambar 4. 37 Pengujian algoritma dengan data bus

Pada gambar 4.34 menunjukkan bahwa terdapat 4 bus pada lajur kiri dan 3 bus pada lajur kanan. Total beban bus yang melewati jembatan adalah 198.000 kg sehingga beban kedua jalur melebihi kapasitas maksimal jembatan sehingga portal tertutup. Berikut informasi kendaraan yang ada pada website:



Gambar 4. 38 Tampilan informasi kendaraan pada lajur kanan

Pada gambar 4.38 menunjukkan bahwa hanya ada 1 jenis kendaraan yaitu jenis bus dengan total beban bus pada lajur kanan adalah 85.000 kg.



Gambar 4. 39 Tampilan informasi kendaraan pada lajur kiri

Pada gambar 4.39 menunjukkan bahwa hanya ada 1 jenis kendaraan yaitu jenis bus dengan total beban bus pada lajur kiri adalah 113.000 kg.

4.2 Validasi Algoritma

Validasi algoritma adalah cara untuk memvalidasi apakah algoritma yang dibuat sudah memenuhi standart algoritma yang sudah ada. Dengan dilakukukanya validasi ini dapat mengetahui perbedaan antara algortima yang telah dibuat dengan algoritma yang sudah ada. Beberapa algoritma standart yang dijadikan perbandingan adalah arlogitma mengenai penjadwalan. Penjadwalan adalah sebuah proses pengambilan keputusan yang banyak digunakan oleh perusahaan dan industri. Penjadwalan mengatur banyaknya sumber daya untuk bekerja menghasilkan manfaat yang optimal (tidak banyak menganggur dan tidak banyak antrian). Penjadwalan CPU adalah proses dimana menentukan antrian yang siap untuk dieksekusi oleh CPU (Pinedo, 2016). Terdapat beberapa algoritma penjadwalan CPU diantaranya:

4.2.1 Algoritma penjadwalan First Come First Served (FCFS)

Proses yang pertama kali meminta jatah waktu untuk menggunakan CPU akan dilayani terlebih dahulu. Metode FCFS ini adalah metode yang umum digunakan untuk menyelesaikan permasalahan antrian publik (Platz dan Osterdal, 2015). Contoh proses dalam komputer beserta waktu pelayanannya ditunjukkan pada Tabel 4.1 dibawah ini :

Tabel 4. 1 Daftar Proses Antrian FCFS

Proses	Brust Time (detik)
P1	24
P2	3
P3	3

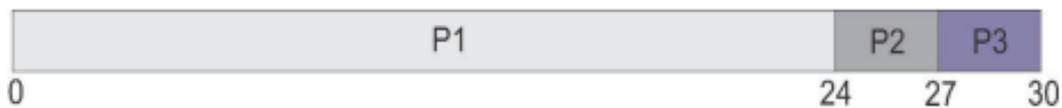
Keterangan :

P1= Proses ke-1 masuk

P2= Proses ke-2 masuk

P3= Proses ke-3 masuk

Kemudian proses antrian akan dilayani / dieksekusi dengan algoritma first come first served. Proses dapat dilihat pada gant chart yang ditunjukkan pada gambar 4.55 :



Gambar 4. 40 Algoritma first come first served.

Gambar 4.40 menunjukkan gant chart dari proses FCFS, proses P1 datang terlebih dahulu dengan beban proses sepanjang 24. Langkah kedua, proses P2 yang datang setelah P1 akan dilayani setelah P1 selesai dilayani. Proses ini akan terus dilakukan hingga semua proses selesai. Penerapannya dalam sistem ini adalah pengunjung yang datang terlebih dahulu dan meminta pelayanan secara langsung akan mendapat jatah pelayanan terlebih dahulu.

Analogi proses antara algoritma yang telah dibuat dengan algoritma FCFS dapat dilihat pada tabel 4.2 .

Tabel 4. 2 Algoritma FCFS dan Algoritma penelitian pada saat pembacaan berat kendaraan oleh loadcell

Algoritma pada penelitian			Algoritma FCFS	
Jenis kendaraan	Posisi urutan kendaraan masuk	Berat kendaraan (kg)	Proses Eksekusi Data	<i>Brush time</i> (detik)
Daihatsu sigra	1	1000	P1 Masuk	3
colt diesel	2	7500	P2 Masuk	4
Mobilio	3	1100	P3 Masuk	5
Daihatsu	4	1540	P4 Masuk	3
Avanza	5	1045	P5 Masuk	5

Kemudian dibandingkan dengan algoritma yang telah dibuat dalam penelitian ini adalah ketika kendaraan mulai memasuki jembatan, maka load cell akan membaca berat kendaraan tersebut. Sebelum proses pembacaan berat kendaraan tersebut selesai maka tidak boleh ada data berat kendaraan lain yang masuk kedalam sistem. Setelah loadcell membaca berat kendaraan, maka sensor akselerometer yang berada di titik sepanjang bentang jembatan akan membaca posisi kendaraan. Pada saat pembacaan posisi kendaraan, sistem kerja yang digunakan sama seperti pembacaan berat kendaraan. Data yang pertama kali di dapat adalah data yang pertama kali masuk.

Tabel 4. 3 Proses pembacaan posisi kendaraan dengan sensor akselerometer saat jembatan dalam kondisi jembatan macet.

Algoritma pada penelitian			Algoritma FCFS	
Jenis kendaraan	Posisi urutan kendaraan masuk	Posisi urutan pada sensor akselerometer	Proses Eksekusi Data	<i>Brush time</i> (detik)
Daihatsu sigra	1	5	P1 Masuk	3
colt diesel double(bak)	2	4	P2 Masuk	4

Lanjutan tabel 4.3

Algoritma pada penelitian			Algoritma FCFS	
Jenis kendaraan	Posisi urutan kendaraan masuk	Posisi urutan pada sensor akselerometer	Proses Eksekusi Data	<i>Brush time</i> (detik)
Honda Mobilio	3	3	P3 Masuk	5
Daihatsu Xenia	4	2	P4 Masuk	3
Avanza	5	1	P5 Masuk	5

Pada saat kondisi macet, maka kendaraan yang masuk pertama kali akan keluar untuk pertama kalinya juga. Hal ini disebabkan karena kondisi jembatan dalam keadaan ramai dan padat sehingga kendaraan yang berada dibelakang tidak bisa mendahului kendaraan yang berada di depannya. Sehingga sensor akselerometer yang berada di titik sepanjang bentang jembatan akan aktif.

Tabel 4. 4 Proses pembacaan posisi kendaraan dengan sensor akselerometer saat jembatan dalam kondisi jembatan macet

Algoritma pada penelitian			Algoritma FCFS	
Jenis kendaraan	Posisi urutan kendaraan masuk	Posisi urutan pada sensor akselerometer	Proses Eksekusi Data	<i>Brush time</i> (detik)
Daihatsu sigra	1	5	P1 Masuk	3
colt diesel double(bak)	2	2	P2 Masuk	4
Honda Mobilio	3	1	P3 Masuk	5
Daihatsu Xenia	4	3	P4 Masuk	3
Avanza	5	4	P5 Masuk	5

4.2.2 Algoritma penjadwalan Round Robin.

Algoritma Round Robin dirancang untuk sistem time sharing. Algoritma ini mirip dengan penjadwal FCFS, namun preemption ditambahkan untuk switch antar proses. Antrian ready diperlakukan atau dianggap sebagai antrian sirkular. CPU mengelilingi antrian ready dan mengalokasikan masing-masing proses untuk *interval* waktu tertentu sampai satu *time slice* atau *quantum*. Berdasarkan beberapa algoritma penjadualan, algoritma yang mungkin diterapkan untuk antrian manusia dan untuk sistem ini adalah algoritma FCFS dan SJF. Contoh proses dalam komputer beserta waktu pelayanannya ditunjukkan pada tabel 4.5 :

Tabel 4. 5 Proses dalam komputer beserta waktu pelayanan

Proses	Burst Time (detik)
P1	50
P2	15
P3	30

(Halaman ini sengaja dikosongkan)

BAB 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Adapun kesimpulan dari thesis ini adalah :

1. Algoritma yang dirancang pada sistem monitoring dan kontrol jembatan ini adalah algoritma untuk mengatur kendaraan agar kendaraan yang memasuki jembatan tidak melebihi kapasitas beban maksimal jembatan. Dalam tesis ini diasumsikan kapasitas maksimal jembatan adalah 100.000 kg. Kendaraan yang masuk jembatan telah di index oleh beberapa variabel dalam kurun waktu 24 jam. Beberapa variabel tersebut adalah waktu, identitas kendaraan berupa plat nomer, jenis kendaraan, berat kendaraan dan index. Dalam menguji kebenaran algoritma maka digunakan perangkat lunak *visual basic*. Data kendaraan riil di persiapkan dalam data excel yang dapat mewakili data sebenarnya. Algoritma kontrol pada jembatan yang diterapkan adalah ketika kendaraan memasuki jembatan maka sensor loadcell akan mendeteksi berat kendaraan lalu sensor akselerometer mendeteksi posisi kendaraan. Data kendaraan yang melewati akan tersimpan dalam buffer dan akan dijumlahkan. Apabila kendaraan melewati sensor akselerometer ke 8 maka data kendaraan tersebut akan terhapus. Sehingga data yang tersimpan hanyalah data kendaraan yang masih ada diatas jembatan. Eksekusi penutupan portal berdasarkan hasil perhitungan dari beban kendaraan yang terukur. Apabila beban kendaraan melebihi kapasitas beban maksimal jembatan yang ditetapkan maka portal akan menutup jembatan dan sebaliknya. Informasi mengenai kondisi jembatan ditampilkan melalui website berbasis *Internet of Things (IoT)*.
2. Memvalidasi perancangan algoritma sistem monitoring dan kontrol jembatan berbasis *Internet Of Things* secara *realtime* ini adalah dengan membandingkannya dengan algoritma standart yang sudah ada. Algoritma standart yang diterapkan adalah algoritma *first come first serve* yang menjadi prinsip dalam pembuatan langkah-langkah program. Sehingga data yang pertama di eksekusi adalah data yang pertama kali

disimpan. Dalam penelitian ini sensor load cell akan membaca data berat kendaraan sesuai dengan kendaraan pertama yang melewatinya. Sedangkan sensor akselerometer akan mendeteksi keberadaan kendaraan baik dalam kondisi macet, lengang dan normal. Sehingga setiap data yang masuk dieksekusi terlebih dahulu.

5.2 Saran

Adapun saran dari thesis ini adalah:

1. Realisasi pembuatan *prototype* sistem monitoring dan kontrol jembatan lebih baik dilakukan agar pembuktian kebenaran algoritma dapat lebih terukti.

DAFTAR PUSTAKA

- Effendi, M., & Hendarto, R. M. (2014). Dampak Pembangunan Jembatan Suramadu Terhadap Perekonomian Pulau Madura (Studi Kasus Kabupaten Bangkalan). *Diponegoro Journal of Economics*, 3(1), 1–13. Retrieved from <http://ejournal-s1.undip.ac.id/index.php/jme>
- Egisa Tarwina Maris, E. Y. H. (2018). Studi Deformasi Jembatan Suramadu Akibat Pengaruh Traffic Load. *Journal of Chemical Information and Modeling*, 53(9), 1689–1699. <https://doi.org/10.1017/CBO9781107415324.004>
- Putra, S. A., Sani, G. A. A., Nurwijaya, A. T., Anandadiga, A., Wijayanto, P. B., Trilaksono, B. R., & Riyansyah, M. (2018). Sistem Penilaian Kondisi Jembatan Menggunakan Respons Dinamik dengan Wireless Sensor Network. *Jurnal Nasional Teknik Elektro Dan Teknologi Informasi (JNTETI)*, 7(3), 338–343. <https://doi.org/10.22146/jnteti.v7i3.444>
- Syaryadhi, M., Hasibuan, P., & Suhardi, S. (2016). Penggunaan Accelerometer MMA7361 sebagai Alternatif Pengukuran Lendutan pada Jembatan Secara Nirkabel Berbasis ATmega32. *Jurnal Rekayasa Elektrika*, 11(5), 183. <https://doi.org/10.17529/jre.v11i5.3215>
- Widjajakusumadan, J., Wiliany, F., Sipil, J. T., Harapan, U. P., & Karawaci, L. (2013). *Vibrasi sistem dengan derajat kebebasan satu $\zeta \omega \omega$* . 7(KoNTekS 7), 24–26.
- Hasan, Y. (2012). *Implementasi Algoritma Multilevel Feedback Queue Dalam Menentukan Waktu Tunggu Dan Waktu Diterbitkan Oleh : STMIK Budi Darma Medan Diterbitkan Oleh : STMIK Budi Darma Medan*. II(June), 51–56.
- Rahadi, T. (2015). *Klasifikasi Jembatan*. Universitas Sumatera Utara.
- Rizal Adhi Pratama. (2013). Pengamatan Lendutan Vertikal Jembatan Kali Babon Dengan Metode Terrestrial Laser Scanner. *Jurnal Geodesi Undip*, 2(4), 279–293.
- Ju, S. H. (2013). Improvement of bridge structures to increase the safety of moving trains during earthquakes. *Engineering Structures*, 56, 501–508. <https://doi.org/10.1016/j.engstruct.2013.05.035>

- Zhao, J., Liu, T., & Wang, Y. (2011). Static Test Analysis of a Bridge Structure in Civil Engineering. *Systems Engineering Procedia*, 1, 10–15. <https://doi.org/10.1016/j.sepro.2011.08.003>
- Satya Wacana Christian. (2016). The Application of Fishbone Diagram Analisis to Improve School Quality. *Dinamika Ilmu*, 16(1), 59–74.
- Adanur, S., Altunişik, A. C., Soyluk, K., Bayraktar, A., & Dumanoglu, A. A. (2016). Multiple-support seismic response of Bosphorus Suspension Bridge for various random vibration methods. *Case Studies in Structural Engineering*, 5, 54–67. <https://doi.org/10.1016/j.csse.2016.04.001>
- Effendi, M. (2013). *DAMPAK PEMBANGUNAN JEMBATAN SURAMADU TERHADAP PEREKONOMIAN PULAU MADURA (Studi Kasus Kabupaten Bangkalan) SKRIPSI*. 14–58.
- Effendi, M., & Hendarto, R. M. (2014). Dampak Pembangunan Jembatan Suramadu Terhadap Perekonomian Pulau Madura (Studi Kasus Kabupaten Bangkalan). *Diponegoro Journal of Economics*, 3(1), 1–13. Retrieved from <http://ejournal-s1.undip.ac.id/index.php/jme>
- Fitzgerald, P. C., Sevillano, E., Obrien, E. J., & Malekjafarian, A. (2017). Bridge weigh-in-motion using a moving force identification algorithm. *Procedia Engineering*, 199, 2955–2960. <https://doi.org/10.1016/j.proeng.2017.09.429>
- Fujino, Y., Siringoringo, D. M., Ikeda, Y., Nagayama, T., & Mizutani, T. (2019). Research and Implementations of Structural Monitoring for Bridges and Buildings in Japan. *Engineering*, (xxxx). <https://doi.org/10.1016/j.eng.2019.09.006>
- Gatti, M. (2019). Structural health monitoring of an operational bridge: A case study. *Engineering Structures*, 195(November 2018), 200–209. <https://doi.org/10.1016/j.engstruct.2019.05.102>
- Girona, M., & Aparicio, Á. C. (2016). Active control implementation in cable-stayed bridges for quasi-static loading patterns. *Engineering Structures*, 118,

394–406. <https://doi.org/10.1016/j.engstruct.2016.03.061>

Kwon, H. C., Kim, M. C., & Lee, I. W. (1998). Vibration control of bridges under moving loads. *Computers and Structures*, 66(4), 473–480. [https://doi.org/10.1016/S0045-7949\(97\)00087-4](https://doi.org/10.1016/S0045-7949(97)00087-4)

Leander, J., Andersson, A., & Karoumi, R. (2010). Monitoring and enhanced fatigue evaluation of a steel railway bridge. *Engineering Structures*, 32(3), 854–863. <https://doi.org/10.1016/j.engstruct.2009.12.011>

Li, W., & Kara, S. (2017). Methodology for Monitoring Manufacturing Environment by Using Wireless Sensor Networks (WSN) and the Internet of Things (IoT). *Procedia CIRP*, 61, 323–328. <https://doi.org/10.1016/j.procir.2016.11.182>

Maris, E. (ITS). (2018). studi deformasi jembatan suramadu akibat pengaruh traffic load. *Journal of Chemical Information and Modeling* Egisa Tarwina Maris, E. Y. H. (2018). Studi Deformasi Jembatan Suramadu Akibat Pengaruh Traffic Load. *Journal of Chemical Information and Modeling*, 53(9), 1689–1699. <https://doi.org/10.1017/CBO9781107415324.004>, 53(9), 1689–1699. <https://doi.org/10.1017/CBO9781107415324.004>

Mei, Q., Gül, M., & Boay, M. (2019). Indirect health monitoring of bridges using Mel-frequency cepstral coefficients and principal component analysis. *Mechanical Systems and Signal Processing*, 119, 523–546. <https://doi.org/10.1016/j.ymsp.2018.10.006>

Meutia, E. D. (2015). Internet of Things – Keamanan dan Privasi. *Seminar Nasional Dan Expo Teknik Elektro 2015*, 85–89.

Moulat, M. El, Debauche, O., Mahmoudi, S., Brahim, L. A., Manneback, P., & Lebeau, F. (2018). Monitoring System Using Internet of Things for Potential Landslides. *Procedia Computer Science*, 134, 26–34. <https://doi.org/10.1016/j.procs.2018.07.140>

Putra, S. A., Sani, G. A. A., Nurwijaya, A. T., Anandadiga, A., Wijayanto, P. B.,

- Trilaksono, B. R., & Riyansyah, M. (2018). Sistem Penilaian Kondisi Jembatan Menggunakan Respons Dinamik dengan Wireless Sensor Network. *Jurnal Nasional Teknik Elektro Dan Teknologi Informasi (JNTETI)*, 7(3), 338–343. <https://doi.org/10.22146/jnteti.v7i3.444>
- Rahadi, T. (2015). *Klasifikasi Jembatan*. Universitas Sumatera Utara.
- Rizal Adhi Pratama. (2013). Pengamatan Lendutan Vertikal Jembatan Kali Babon Dengan Metode Terrestrial Laser Scanner. *Jurnal Geodesi Undip*, 2(4), 279–293.
- Satya Wacana Christian. (2016). The Application of Fishbone Diagram Analisis to Improve School Quality. *Dinamika Ilmu*, 16(1), 59–74.
- Syaryadhi, M. (2016). Penggunaan Accelerometer MMA7361 sebagai Alternatif Pengukuran Lendutan pada Jembatan Secara Nirkabel Berbasis ATmega32. *Jurnal Rekayasa Elektrika*, 11(5), 183. <https://doi.org/10.17529/jre.v11i5.3215>
- Tylee, L. (1998). Learn Visual Basic 6 . 0. *Distribution*, 98008(206).
- Yu, Y., Zhao, X., Shi, Y., & Ou, J. (2013). Design of a real-time overload monitoring system for bridges and roads based on structural response. *Measurement: Journal of the International Measurement Confederation*, 46(1), 345–352. <https://doi.org/10.1016/j.measurement.2012.07.006>
- Zhao, J., Liu, T., & Wang, Y. (2011). Static Test Analysis of a Bridge Structure in Civil Engineering. *Systems Engineering Procedia*, 1, 10–15. <https://doi.org/10.1016/j.sepro.2011.08.003>

Lampiran

Lampiran 1

Data informasi kendaraan untuk lajur kanan

No.	Nama Kendaraan		Jenis Kendaraan	Plat Nomer		Berat Kendaraan (Kg)
	Brand	Type				
1	Daihatsu sigra	1500 CC	Kendaraan Pribadi	L	2314	1000
2	colt diesel double(bak)	3908 cc	Truk	L	4231	7500
3	Honda Mobilio	1500 CC	Kendaraan Pribadi	M	1313	1100
4	Daihatsu Xenia	Mi 1000 CC	Kendaraan Pribadi	S	2416	1540
5	Avanza	1500 CC	Kendaraan Pribadi	L	2131	1045
6	colt diesel double	3908 cc	Truk	L	1312	8000
7	Heavy dum truk	3908 cc	Truk	L	8492	8000
8	Expander	1500 CC	Kendaraan Pribadi	M	5674	1780
9	colt diesel double(Box)	3908 cc	Truk	S	7864	8000
10	Toyota Alpard	2400 cc	Kendaraan Pribadi	L	4544	1910
11	Yamaha R25	250 cc	Kendaraan Pribadi	L	6790	267
12	Truck Wing Box Tronton	7500 cc	Truk	L	3468	18000
13	Honda Vario 150 eSP	150 cc	Kendaraan Pribadi	M	9675	215
14	Honda Scoopy F1	110 cc	Kendaraan Pribadi	S	5799	170
15	bus(60 orang)	7648 c	Bus	L	9987	30000

Lanjutan lampiran 1

No.	Nama Kendaraan		Jenis Kendaraan	Plat Nomer		Berat Kendaraan (Kg)
	Brand	Type				
16	Kijang Innova	2000cc	Kendaraan Pribadi	L	8975	1860
17	Toyota Fortuner	2694 cc	Kendaraan Pribadi	L	5664	1800
18	Toyota NAV1	1000 cc	Kendaraan Pribadi	M	2343	800
19	Yahama R15 V3	155 cc	Kendaraan Pribadi	S	6864	243
20	colt diesel double(motor carrier long)	7545 cc	Truk	L	7787	14000
21	Honda ADV 150	150 cc	Kendaraan Pribadi	L	6543	210
22	colt diesel double(car carrier)	3908 cc	Truk	L	9097	8000
23	Honda Beat 2019	110 cc	Kendaraan Pribadi	M	7634	160
24	New Yaris	1497 cc	Kendaraan Pribadi	S	7645	1045
25	Yamaha N Max	155 cc	Kendaraan Pribadi	L	6677	231
26	colt diesel double(bak long)	7545 cc	Truk	L	8896	14000
27	Grand New Avanza Veloz	1329 cc	Kendaraan Pribadi	L	9875	1085
28	honda vario	125 cc	Kendaraan Pribadi	M	4577	200
29	colt diesel double(long box)	7545 cc	Truk	S	785	14000
30	colt diesel double(los bak)	3908 cc	Truk	L	9954	8000

Lampiran 2

Data informasi kendaraan untuk lajur kiri

No.	Nama Kendaraan		Jenis Kendaraan	Plat Nomer		Berat Kendaraan (Kg)
	Brand	Type				
1	colt diesel double	3908 cc	Truk	S	2121	8000
2	Toyota Alpard	2400 cc	Kendaraan Pribadi	L	4241	1910
3	Yamaha R25	250 cc	Kendaraan Pribadi	L	2131	267
4	Truck Wing Box Tronton	7500 cc	Truk	L	3111	18000
5	Honda Vario 150 eSP	150 cc	Kendaraan Pribadi	M	5343	215
6	Honda Scoopy F1	110 cc	Kendaraan Pribadi	S	2313	170
7	bus(60 orang)	7648 c	Bus	L	6765	30000
8	Kijang Innova Reborn	2000cc	Kendaraan Pribadi	L	9855	1860
9	Toyota Fortuner	2694 cc	pribadi	L	4646	1800
10	Toyota NAV1	1000 cc	Kendaraan Pribadi	M	7464	800
11	Yahama R15 V3	155 cc	Kendaraan Pribadi	S	7453	243
12	colt diesel)	7545 cc	Truk	L	5635	14000
13	Daihatsu sigra	1500 CC	Kendaraan Pribadi	L	6252	1000
14	colt diesel	3908 cc	Truk	L	5222	7500
15	Honda Mobilio	1500 CC	Kendaraan Pribadi	M	3121	1100

Lanjutan lampiran 2

No.	Nama Kendaraan		Jenis Kendaraan	Plat Nomer		Berat Kendaraan (Kg)
	Brand	Type				
16	Daihatsu Xenia	Mi 1000 CC	Kendaraan Pribadi	S	3131	1540
17	Avanza	1500 CC	Kendaraan Pribadi	L	2131	1045
18	colt diesel double	3908 cc	Truk	L	1213	8000
19	Heavy dum truk	3908 cc	Truk	L	8492	8000
20	Expander	1500 CC	Kendaraan Pribadi	M	6744	1780
21	Honda ADV 150	150 cc	Kendaraan Pribadi	L	5342	210
22	colt diesel double	3908 cc	Truk	L	2414	8000
23	Honda Beat 2019	110 cc	Kendaraan Pribadi	M	7634	160
24	New Yaris	1497 cc	Kendaraan Pribadi	S	6342	1045
25	Yamaha N Max	155 cc	Kendaraan Pribadi	L	6677	231
26	colt diesel	7545 cc	Truk	L	5231	14000
27	Grand New Avanza Veloz	1329 cc	Kendaraan Pribadi	L	9875	1085
28	honda vario	125 cc	Kendaraan Pribadi	M	5466	200
29	colt diesel double(long box)	7545 cc	Truk	S	9675	14000
30	colt diesel double(los bak)	3908 cc	Truk	L	6342	8000

Lampiran 3

Program untuk algoritma kontrol pada jembatan menggunakan perangkat lunak *visual basic*.

```
Imports Excel = Microsoft.Office.Interop.Excel

Imports MySql.Data.MySqlClient

Public Class Main

Dim conn As MySqlConnection = New
MySqlConnection("server=localhost;userid=root;password=;database=jembatan")

    Dim jmldt As Integer = 30

    Dim a As Integer = 0

    Dim b As Integer = 0

    Dim APP As New Excel.Application

    Dim worksheet As Excel.Worksheet

    Dim workbook As Excel.Workbook

    Dim jenis(jmldt - 1) As String

    Dim jenis2(jmldt - 1) As String

    Dim arr1(jmldt - 1) As Integer

    Dim arr2(jmldt - 1) As Integer

    Dim vhc1 As Integer = 0

    Dim vhc2 As Integer = 0

    Dim total1 As Integer = 0

    Dim total2 As Integer = 0

    Dim truk As Integer = 0

    Dim bus As Integer = 0

    Dim kp As Integer = 0

    Dim truk2 As Integer = 0

    Dim bus2 As Integer = 0

    Dim kp2 As Integer = 0

    Dim v1g, v2g As PictureBox()
```

```

Dim maxload As Integer = 300000

Dim sum, intNumber, nKondisi1, nKondisi2 As Integer

Dim arrNumber1(jmltdt - 1) As Integer

Dim arrNumber2(jmltdt - 1) As Integer

Private Sub Main_Load(sender As Object, e As EventArgs) Handles MyBase.Load

    workbook = APP.Workbooks.Open("F:\AKADEMIK S2\SEMESTER 3\Bismillah program
    fix\30 kendaraan.xlsx")

    worksheet = workbook.Worksheets("Sheet2")

    Try

        conn.Open()

        Dim addquery As String = "UPDATE `record` SET `Jumlah Kendaraan` = '0' WHERE
        `record`.`No` = 1;"

        Dim cmd As MySqlCommand = New MySqlCommand(addquery, conn)

        cmd.ExecuteNonQuery()

        addquery = "UPDATE `record` SET `Jumlah Kendaraan` = '0' WHERE `record`.`No` = 2;"

        cmd = New MySqlCommand(addquery, conn)

        cmd.ExecuteNonQuery()

        addquery = "UPDATE `record` SET `Jumlah Kendaraan` = '0' WHERE `record`.`No` = 3;"

        cmd = New MySqlCommand(addquery, conn)

        cmd.ExecuteNonQuery()

        addquery = "UPDATE `weight` SET `beban` = '0' WHERE `weight`.`No` = 1;"

        cmd = New MySqlCommand(addquery, conn)

        cmd.ExecuteNonQuery()

        addquery = "UPDATE `record` SET `Jumlah Kendaraan 2` = '0' WHERE `record`.`No` =
        1;"

        cmd = New MySqlCommand(addquery, conn)

        cmd.ExecuteNonQuery()

        addquery = "UPDATE `record` SET `Jumlah Kendaraan 2` = '0' WHERE `record`.`No` =
        2;"

        cmd = New MySqlCommand(addquery, conn)

        cmd.ExecuteNonQuery()
    
```

```

    addquery = "UPDATE `record` SET `Jumlah Kendaraan 2` = '0' WHERE `record`.`No` = 3;"

    cmd = New MySqlCommand(addquery, conn)

    cmd.ExecuteNonQuery()

    addquery = "UPDATE `weight` SET `beban` = '0' WHERE `weight`.`No` = 2;"

    cmd = New MySqlCommand(addquery, conn)

    cmd.ExecuteNonQuery()

Catch ex As Exception

    MessageBox.Show("Cannot connect to database: " & ex.Message)

Finally

    conn.Dispose()

End Try

End Sub

Sub kondisi1(m As Integer, n As Integer)

    total1 = 0

    truk = 0

    bus = 0

    kp = 0

    nKondisi1 = Int((m * Rnd()) + n)

    Dim bantu(30) As Integer

    Dim bantu_jns(30) As String

    randarr1(nKondisi1)

    For g = 0 To nKondisi1 - 1

        bantu(g) = arr1(arrNumber1(g))

        bantu_jns(g) = jenis(arrNumber1(g))

        If bantu_jns(g) = "Truk" Then

            truk = truk + 1

        ElseIf bantu_jns(g) = "Bus" Then

```

```

        bus = bus + 1

    ElseIf bantu_jns(g) = "Kendaraan Pribadi " Then

        kp = kp + 1

    End If

    total1 = total1 + bantu(g)

Next

For h = 0 To jmltdt - 1

    arr1(h) = bantu(h)

    jenis(h) = bantu_jns(h)

Next

LabelBerat1.Text = total1.ToString

vhc1 = truk + bus + kp

LabelKendaraan1.Text = vhc1.ToString

End Sub

Sub kondisi2(m As Integer, n As Integer)

    total2 = 0

    truk2 = 0

    bus2 = 0

    kp2 = 0

    nKondisi2 = Int((m * Rnd()) + n)

    Dim bantu2(30) As Integer

    Dim bantu_jns2(30) As String

    randarr2(nKondisi2)

    For g = 0 To nKondisi2 - 1

        bantu2(g) = arr2(arrNumber2(g))

        bantu_jns2(g) = jenis2(arrNumber2(g))

        If bantu_jns2(g) = "Truk" Then

            truk2 = truk2 + 1

        ElseIf bantu_jns2(g) = "Bus" Then

```

```

        bus2 = bus2 + 1

    ElseIf bantu_jns2(g) = "Kendaraan Pribadi " Then

        kp2 = kp2 + 1

    End If

    total2 = total2 + bantu2(g)

Next

For h = 0 To jmltdt - 1

    arr2(h) = bantu2(h)

    jenis2(h) = bantu_jns2(h)

Next

LabelBerat2.Text = total2.ToString

vhc2 = truk2 + bus2 + kp2

LabelKendaraan2.Text = vhc2.ToString

End Sub

Private Sub LC1_1_Click(sender As Object, e As EventArgs) Handles LC1_1.Click

    total1 = 0

    truk = 0

    bus = 0

    kp = 0

    For y = 0 To jmltdt - 1

        jenis(y) = worksheet.Cells(y + 4, 4).Value

        Dim jns1 As String = worksheet.Cells(y + 4, 4).Value

        If jns1 = "Truk" Then

            truk = truk + 1

        ElseIf jns1 = "Bus" Then

            bus = bus + 1

        ElseIf jns1 = "Kendaraan Pribadi " Then

            kp = kp + 1

        End If
    
```

```

arr1(y) = worksheet.Cells(y + 4, 7).Value '4,7

total1 = total1 + arr1(y)

Next

LabelBerat1.Text = total1.ToString

vhc1 = truk + bus + kp

LabelKendaraan1.Text = vhc1.ToString

If rb_lgg.Checked Then

    kondisi1(6, 5) '5-10vhc

End If

If rb_nrml.Checked Then

    kondisi1(9, 10) '10-18 vhc

End If

Try

    conn.Open()

    Dim addquery As String = "UPDATE `record` SET `Jumlah Kendaraan` = " & truk & "
    WHERE `record`.`No` = 1;"

    Dim cmd As MySqlCommand = New MySqlCommand(addquery, conn)

    cmd.ExecuteNonQuery()

    addquery = "UPDATE `record` SET `Jumlah Kendaraan` = " & kp & " WHERE
    `record`.`No` = 2;"

    cmd = New MySqlCommand(addquery, conn)

    cmd.ExecuteNonQuery()

    addquery = "UPDATE `record` SET `Jumlah Kendaraan` = " & bus & " WHERE
    `record`.`No` = 3;"

    cmd = New MySqlCommand(addquery, conn)

    cmd.ExecuteNonQuery()

    addquery = "UPDATE `weight` SET `beban` = " & total1 & " WHERE `weight`.`No` =
    1;"

    cmd = New MySqlCommand(addquery, conn)

    cmd.ExecuteNonQuery()

```

Catch ex As Exception

 MessageBox.Show("Cannot connect to database: " & ex.Message)

Finally

 MessageBox.Show("Data Berhasil Dimasukkan")

 conn.Dispose()

End Try

palangG1.Show()

Timer2.Enabled = True

End Sub

Private Sub LC2_2_Click(sender As Object, e As EventArgs) Handles LC2_2.Click

 total2 = 0

 truk2 = 0

 bus2 = 0

 kp2 = 0

 For y = 0 To jmltdt - 1

 jenis2(y) = worksheet.Cells(y + 4, 12).Value

 Dim jns2 As String = worksheet.Cells(y + 4, 12).Value

 If jns2 = "Truk" Then

 truk2 = truk2 + 1

 ElseIf jns2 = "Bus" Then

 bus2 = bus2 + 1

 ElseIf jns2 = "Kendaraan Pribadi " Then

 kp2 = kp2 + 1

 End If

 arr2(y) = worksheet.Cells(y + 4, 15).Value '4,15

 total2 = total2 + arr2(y)

 Next

LabelBerat2.Text = total2.ToString

vhc2 = truk2 + bus2 + kp2

```
LabelKendaraan2.Text = vhc2.ToString
```

```
If rb_lgg2.Checked Then
```

```
    kondisi2(6, 5) '5-10vhc
```

```
End If
```

```
If rb_nrm12.Checked Then
```

```
    kondisi2(9, 10) '10-18 vhc
```

```
End If
```

```
Try
```

```
    conn.Open()
```

```
    Dim addquery As String = "UPDATE `record` SET `Jumlah Kendaraan 2` = " & truk2 & "" WHERE `record`.`No` = 1;"
```

```
    Dim cmd As MySqlCommand = New MySqlCommand(addquery, conn)
```

```
    cmd.ExecuteNonQuery()
```

```
    addquery = "UPDATE `record` SET `Jumlah Kendaraan 2` = " & kp2 & "" WHERE `record`.`No` = 2;"
```

```
    cmd = New MySqlCommand(addquery, conn)
```

```
    cmd.ExecuteNonQuery()
```

```
    addquery = "UPDATE `record` SET `Jumlah Kendaraan 2` = " & bus2 & "" WHERE `record`.`No` = 3;"
```

```
    cmd = New MySqlCommand(addquery, conn)
```

```
    cmd.ExecuteNonQuery()
```

```
    addquery = "UPDATE `weight` SET `beban` = " & total2 & "" WHERE `weight`.`No` = 2;"
```

```
    cmd = New MySqlCommand(addquery, conn)
```

```
    cmd.ExecuteNonQuery()
```

```
Catch ex As Exception
```

```
    MessageBox.Show("Cannot connect to database: " & ex.Message)
```

```
Finally
```

```
    MessageBox.Show("Data Berhasil Dimasukkan")
```

```
    conn.Dispose()
```

```

End Try

palangG2.Show()

Timer3.Enabled = True

End Sub

Sub randarr1(r As Integer)

    For c = 0 To jmltd - 1

        arrNumber1(c) = 100

    Next

    Dim i, x, y As Integer

    Label8.Text = "

    For x = 0 To r - 1

Start:

        intNumber = (Int((r * Rnd()) + 1)) - 1

        For y = 0 To r - 1

            If intNumber = arrNumber1(y) Then

                GoTo Start

            End If

        Next y

        arrNumber1(x) = intNumber

    Next x

    For i = 0 To r - 1

        Label8.Text = Label8.Text & (arrNumber1(i)) & ", "

    Next

End Sub

Sub randarr2(r As Integer)

    For c = 0 To jmltd - 1

        arrNumber2(c) = 100

    Next

```

```

Dim i, x, y As Integer

Label9.Text = ""

'-----

For x = 0 To r - 1
Start:
    intNumber = (Int((r * Rnd()) + 1)) - 1
    For y = 0 To r - 1
        If intNumber = arrNumber2(y) Then
            GoTo Start
        End If
    Next y
    arrNumber2(x) = intNumber
Next x
For i = 0 To r - 1
    Label9.Text = Label9.Text & (arrNumber2(i)) & " , "
Next
End Sub

Private Sub V1g8_Click(sender As Object, e As EventArgs) Handles v1g8.Click
    truk = 0
    bus = 0
    kp = 0

    If rb_mct.Checked = True Then
        randarr1(vhc1)
        If vhc1 > 0 Then
            total1 = 0
            If vhc1 = jmltd Then
                For y = 0 To jmltd - 1
                    If y < jmltd - 1 Then

```

```

        arr1(y) = arr1(y + 1)

        jenis(y) = jenis(y + 1)

    Else

        arr1(jmltdt - 1) = 0

        jenis(jmltdt - 1) = ""

    End If

Next

Else

    For y = 0 To jmltdt - 1

        If y < jmltdt - 1 Then

            arr1(y) = arr1(y + 1)

            jenis(y) = jenis(y + 1)

        End If

    Next

End If

For i = 0 To jmltdt - 1

    If jenis(i) = "Truk" Then

        truk = truk + 1

    ElseIf jenis(i) = "Bus" Then

        bus = bus + 1

    ElseIf jenis(i) = "Kendaraan Pribadi " Then

        kp = kp + 1

    End If

    total1 = total1 + arr1(i)

Next

LabelBerat1.Text = total1

vhc1 = vhc1 - 1

LabelKendaraan1.Text = vhc1

```

```

End If

ElseIf rb_nrml.Checked = True Or rb_lgg.Checked Then

    arr1(arrNumber1(vhc1 - 1)) = 0

    jenis(arrNumber1(vhc1 - 1)) = ""

    total1 = 0

    For i = 0 To jmldt - 1

        If jenis(i) = "Truk" Then

            truk = truk + 1

        ElseIf jenis(i) = "Bus" Then

            bus = bus + 1

        ElseIf jenis(i) = "Kendaraan Pribadi " Then

            kp = kp + 1

        End If

        total1 = total1 + arr1(i)

    Next

    LabelBerat1.Text = total1

    vhc1 = vhc1 - 1

    LabelKendaraan1.Text = vhc1

End If

Try

    conn.Open()

    Dim addquery As String = "UPDATE `record` SET `Jumlah Kendaraan` = " & truk & "
    WHERE `record`.`No` = 1;"

    Dim cmd As MySqlCommand = New MySqlCommand(addquery, conn)

    cmd.ExecuteNonQuery()

```

```
addquery = "UPDATE `record` SET `Jumlah Kendaraan` = " & kp & " WHERE  
`record`.`No` = 2;"
```

```
cmd = New MySqlCommand(addquery, conn)
```

```
cmd.ExecuteNonQuery()
```

```
addquery = "UPDATE `record` SET `Jumlah Kendaraan` = " & bus & " WHERE  
`record`.`No` = 3;"
```

```
cmd = New MySqlCommand(addquery, conn)
```

```
cmd.ExecuteNonQuery()
```

```
addquery = "UPDATE `weight` SET `beban` = " & total1 & " WHERE `weight`.`No` =  
1;"
```

```
cmd = New MySqlCommand(addquery, conn)
```

```
cmd.ExecuteNonQuery()
```

```
Catch ex As Exception
```

```
    MessageBox.Show("Cannot connect to database: " & ex.Message)
```

```
Finally
```

```
    MessageBox.Show("Kendaraan Keluar")
```

```
    conn.Dispose()
```

```
End Try
```

```
End Sub
```

```
Private Sub V2g1_Click(sender As Object, e As EventArgs) Handles v2g1.Click
```

```
    truk2 = 0
```

```
    bus2 = 0
```

```
    kp2 = 0
```

```
    If rb_mct2.Checked = True Then
```

```
        randarr2(vhc2)
```

```
        If vhc2 > 0 Then
```

```
            total2 = 0
```

```

If vhc2 = jmltd Then
  For y = 0 To jmltd - 1
    If y < jmltd - 1 Then
      arr2(y) = arr2(y + 1)
      jenis2(y) = jenis2(y + 1)
    Else
      arr2(jmltd - 1) = 0
      jenis2(jmltd - 1) = ""
    End If
  Next
Else
  For y = 0 To jmltd - 1
    If y < jmltd - 1 Then
      arr2(y) = arr2(y + 1)
      jenis2(y) = jenis2(y + 1)
    End If
  Next
End If

For i = 0 To jmltd - 1
  If jenis2(i) = "Truk" Then
    truk2 = truk2 + 1
  ElseIf jenis2(i) = "Bus" Then
    bus2 = bus2 + 1
  ElseIf jenis2(i) = "Kendaraan Pribadi " Then
    kp2 = kp2 + 1
  End If
  total2 = total2 + arr2(i)
Next

```

```

LabelBerat2.Text = total2

vhc2 = vhc2 - 1

LabelKendaraan2.Text = vhc2

End If

ElseIf rb_nrml2.Checked = True Or rb_lgg2.Checked Then

arr2(arrNumber2(vhc2 - 1)) = 0

jenis2(arrNumber2(vhc2 - 1)) = ""

total2 = 0

For i = 0 To jmltdt - 1

If jenis2(i) = "Truk" Then

truk2 = truk2 + 1

ElseIf jenis2(i) = "Bus" Then

bus2 = bus2 + 1

ElseIf jenis2(i) = "Kendaraan Pribadi " Then

kp2 = kp2 + 1

End If

total2 = total2 + arr2(i)

Next

LabelBerat2.Text = total2

vhc2 = vhc2 - 1

LabelKendaraan2.Text = vhc2

```

Lampiran 4

Program pada website

```
<html lang="en" class="no-js">
<head>
  <meta charset="UTF-8" />
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Weighing Bridge</title>
  <meta name="description" content="Various styles and inspiration for
responsive, flexbox-based HTML pricing tables" />
  <meta name="keywords" content="pricing table, inspiration, ui, modern,
responsive, flexbox, html, component" />
  <meta name="author" content="Codrops" />
  <link rel="shortcut icon" href="favicon.ico">
  <link href='https://fonts.googleapis.com/css?family=Homemade+Apple'
rel='stylesheet' type='text/css'>
  <link href='https://fonts.googleapis.com/css?family=Sahitya:400,700'
rel='stylesheet' type='text/css'>
  <link href='https://fonts.googleapis.com/css?family=Roboto:400,700'
rel='stylesheet' type='text/css'>
  <link href='https://fonts.googleapis.com/css?family=Playfair+Display:900'
rel='stylesheet' type='text/css'>
  <link
href='https://fonts.googleapis.com/css?family=Alegreya+Sans:400,700,800'
rel='stylesheet' type='text/css'>
  <link
href='https://fonts.googleapis.com/css?family=Roboto+Condensed:400,300,700'
rel='stylesheet' type='text/css'>
  <link href='https://fonts.googleapis.com/css?family=PT+Sans:400,700'
rel='stylesheet' type='text/css'>
```

```

<link href='https://fonts.googleapis.com/css?family=Nunito:400,300,700'
rel='stylesheet' type='text/css'>
<link rel="stylesheet" type="text/css" href="css/normalize.css" />
<link rel="stylesheet" type="text/css" href="css/demo.css" />
<link rel="stylesheet" type="text/css" href="css/icons.css" />
<link rel="stylesheet" type="text/css" href="css/component.css" />
<!--[if IE]>
    <script
src="http://html5shiv.googlecode.com/svn/trunk/html5.js"></script>
    <![endif]-->
</head>

<body>
<div class="container">
    <header class="codrops-header">
        <div class="codrops-links">
            <a class="codrops-icon codrops-icon--prev" ></a>
            <a class="codrops-icon codrops-icon--drop" ></a>
        </div>
        <h1><span>Monitoring</span> Jembatan Timbang</h1>
    </header>

    <section class="pricing-section bg-8">
        <h1 class="pricing-section__title">Monitoring Beban Kendaraan</h1>
        <div class="pricing pricing--tashi">
            <div class="pricing__item">
                <h3 class="pricing__title">Total Beban</h3>
                <p class="pricing__sentence">Total beban yang tercatat oleh
sensor</p>
                <div class="pricing__price">50<span class="pricing__period">
Ton</span></div>

```

```
</div>
<div class="pricing__item">
  <h3 class="pricing__title">Truk</h3>
  <p class="pricing__sentence">Jumlah kendaraan melintas dengan
jenis Truk</p>
  <div class="pricing__price">15<span class="pricing__period">
Truk</span></div>
```

```
</div>
<div class="pricing__item">
  <h3 class="pricing__title">Truk Gandeng</h3>
  <p class="pricing__sentence">Jumlah kendaraan melintas dengan
jenis Truk Gandeng</p>
  <div class="pricing__price">2<span class="pricing__period">
Truk</span></div>
```

```
</div>
<div class="pricing__item">
  <h3 class="pricing__title">Bus</h3>
  <p class="pricing__sentence">Jumlah kendaraan melintas dengan
jenis Bus</p>
  <div class="pricing__price">6<span class="pricing__period">
Bus</span></div>
```

```
</div>
<div class="pricing__item">
  <h3 class="pricing__title">Mobil Pribadi</h3>
  <p class="pricing__sentence">Jumlah kendaraan melintas dengan
jenis Mobil</p>
  <div class="pricing__price">20<span class="pricing__period">
Mobil</span></div>
```

```
</div>
</div>
<div class="pricing pricing--rabten">
  <div class="pricing__item">
    <button class="pricing__action">Perbarui Data</button>
  </div>
</div>
</section>
<!-- /container --
```