

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1. Tinjauan Pustaka

Beberapa peneliti terdahulu untuk menyelesaikan permasalahan lalu lintas berbasis *intelligent transport systems* (ITS) telah dilakukan antara lain:

2.1.1. *Intelligen* Transport Systems (ITS)

Implementasi teknologi ITS berbasis *smartway* untuk menyelesaikan permasalahan umum lalu lintas, teknologi ini terdiri dari beberapa komponen antara lain: kamera, *electric toll correction* (ETC) dan *road side unit* (RSU). ETC dan RSU disebut dengan perangkat *on-board unit* (OBU) perangkat tersebut dipasang pada kendaraan yang berfungsi sebagai perangkat pengirim data antara kendaraan dengan pengelola lalu lintas pada area tertentu yang dapat memetakan kepadatan lalu lintas pada area seluas lebih dari 1000 kilometer dan *support* untuk menyajikan informasi keamanan berkendara. Teknologi ini secara otomatis dapat mengirimkan data ke server pengelola lalu lintas. Teknologi tersebut sulit diterapkan untuk monitoring lalu lintas karena membutuhkan biaya besar, pada dasarnya teknologi tersebut memerlukan infrastruktur yang layak sehingga hanya bisa diterapkan pada negara atau kota yang memiliki persiapan yang baik.

2.1.2. *Advance Transportation Management System* (ATMS)

ATMS merupakan gambaran dan strategi pengembangan untuk menyelesaikan permasalahan lalu lintas khususnya pada permasalahan kepadatan lalu lintas. manajemen lalu lintas dinamis dapat memberikan kontrol secara *realtime* untuk optimasi jaringan lalu lintas, sehingga dapat menghasilkan integrasi antara lalu lintas manajemen dinamis dan pemilihan rute secara otomatis yaitu dengan cara memprediksi karakteristik lalu lintas, memeriksa waktu tenggang dan waktu perjalanan. Penelitian tersebut bertujuan mendukung peningkatan kapasitas lalu lintas, pemilihan rute terbaik dan optimasi jaringan lalu lintas untuk mereduksi waktu perjalanan pengguna lalu lintas dengan membahas metode-metode dan strategi yang berguna untuk mendapatkan informasi lalu lintas yang mengombinasikan ke 4 (empat) bagian yaitu *traffic information*, *traffic assignment*, *traffic optimazion* dan *traffic*

prediction untuk menyimpulkan strategi pengembangan pada implementasi ATMS (Shahgholian dan Gharavian, 2018) (Gartner dan Stamatiadis, 2000).

Perkembangan teknologi ATMS menggunakan sensor *loop detection* mampu mendeteksi kendaraan untuk pengawasan lalu lintas dan membedakan kendaraan berdasarkan kelas antara lain: mobil pribadi, mini bus, truk dan trailer. Metode yang digunakan adalah *fuzzy c-means clustering* (FCM) terbukti mampu melakukan klasifikasi kendaraan berdasarkan dimensi dan kecepatan. Untuk pengembangan dan efektivitas maka diusulkan kombinasi dengan metode berbasis deteksi gambar atau video sehingga pengembangan mudah dilakukan (Javadi dkk., 2018).

Penelitian mengenai sistem monitoring pengawasan lalu lintas menggunakan video masih merupakan tugas berat untuk dideteksi dalam adegan dinamis (Hadi dkk., 2014). Pelacakan kendaraan (obyek bergerak) akhir-akhir ini semakin diminati dan muncul sebagai pembelajaran baru terutama pada proyek video *surveillance* dan pelacakan obyek. Desain pada video *surveillance* menuntut untuk dapat melacak secara otomatis dan *realtime* dan juga mampu untuk mengklasifikasikan obyek bergerak. Metode yang digunakan adalah: ekstrak obyek bergerak, pengenalan obyek, pelacakan dan pengambilan keputusan. Kendala yang dihadapi adalah ketika obyek tersebut bersifat berdekatan atau padat, terdapat bayangan, kemacetan lalu lintas, posisi sudut kamera dan peletakan kamera yang tidak sesuai maka akan mempengaruhi hasil prediksi (Kumar., 2013).

2.1.3. Penerapan Metode YOLO Pada Monitoring Lalu Lintas

Kepadatan lalu lintas di daerah perkotaan membutuhkan sistem analisis yang sangat akurat dan cepat untuk pengawasan antara lain dengan menggunakan metode YOLO untuk menganalisis video dalam pendeteksian obyek bergerak. Hasil menunjukkan bahwa teknologi ini mampu bekerja secara *fast traffic* dengan mendapatkan data secara cepat dan akurat untuk pola kepadatan lalu lintas dan meningkatkan manajemen lalu lintas. Namun penelitian ini masih sebatas analisis dan belum memperhitungkan level kepadatan lalu lintas (Babu dan Anirudh, 2018).

Sistem manajemen lalu lintas dengan menggunakan metode YOLO dan filter korelasi diterapkan untuk menghitung jumlah kendaraan. Sistem ini didesain dengan menggunakan bantuan kamera untuk mendeteksi dan menghasilkan *big*

data. Kesulitan dan tantangan dalam mendeteksi obyek secara otomatis ini terletak pada bayangan obyek, kendaraan padat dan tertutup kendaraan lain, hujan, kabut dan variasi lingkungan. Batasan dan kelemahan dari penelitian tersebut adalah meskipun obyek deteksi dapat bekerja secara *realtime* namun hal ini membutuhkan waktu yang cukup lama. Dalam hal ini disarankan pendeteksian obyek pada satu jalur untuk menghindari kepadatan kendaraan (Asha dan Narasimhadhan, 2018).

Sistem deteksi kepadatan lalu lintas yang berbasis *deep convolutional neural network* (DCNN) dan *you look only once* (YOLO) telah diterapkan pada sistem deteksi kepadatan lalu lintas dengan menggunakan bantuan kamera CCTV sebanyak 121 kamera yang tersebar di jalan raya dan terhubung dengan data grafik radar *wavetronix*. Metode *support vector machine* (SVM) juga digunakan sebagai model untuk klasifikasi dalam menentukan jenis kepadatan lalu lintas. Kelebihannya adalah kepadatan lalu lintas dapat dikenali tidak hanya dalam kondisi siang dan malam tetapi juga pada kondisi resolusi kamera baik dan buruk. Metode YOLO meraih akurasi tertinggi yaitu 91.2% diikuti oleh DCNN 90.2% dan 85% *image* berhasil dikoreksi oleh metode SVM. Penelitian tersebut memiliki permasalahan sulit untuk mengurangi *noise* pada grafik sensor *wavetronix* yang mana mempengaruhi deteksi kepadatan lalu lintas (Chakraborty dkk., 2018).

2.1.4. *Euclidean Distance* dan *Pin-Hole* model untuk mengukur kecepatan kendaraan

Pengukuran kecepatan kendaraan dalam video lalu lintas membutuhkan beberapa proses diantaranya: ekstrak video ke dalam bentuk gambar, mendeteksi objek pada gambar dan estimasi kecepatan (Sina, dkk. 2013). Teknik *euclidean distance* adalah teknik pengukuran jarak antara dua titik, secara teknis video terdiri dari gambar-gambar yang menyusun setiap pergerakan objek dalam video, deteksi objek kendaraan dilakukan pada gambar-gambar berseri tersebut kemudian diterapkan ID dan titik *centroid* pada setiap objek yang terdeteksi. Pengukuran antara dua buah titik koordinat centroid dilakukan dengan cara membandingkan titik centroid pada gambar pertama dan gambar berikutnya. Sehingga diperoleh persamaan 2.1.

$$S_{vehicle} = \frac{d_{coordinate} * k}{f_2 - f_1} \quad (2.1)$$

$d_{coordinate}$ merupakan koordinat titik centroid, k adalah *frame rate* pada video biasanya diukur menggunakan satuan *frame per second* (FPS), f_2 adalah *frame rate* saat ini dan f_1 adalah *frame rate* sebelumnya.

Untuk meningkatkan akurasi perhitungan kecepatan kendaraan menggunakan kamera, teknik *pin-hole* model digunakan untuk memperkirakan estimasi jarak sebenarnya dari objek dalam urutan gambar. Ada beberapa parameter yang digunakan pada teknik *pin-hole* model yaitu: perkiraan jarak kamera dengan tanah dan jarak proyeksi kamera dengan memproyeksikan berupa grid pada gambar dengan ukuran mili meter (mm).

Penelitian tersebut menghasilkan kalkulasi kecepatan kendaraan yang diimplementasikan pada malam hari dengan error terendah 3.325 Km/jam.

2.1.5. Perhitungan Dan Simulasi Area *Occupancy*

Pengukuran kepadatan lalu lintas heterogen sulit dilakukan dibandingkan arus lalu lintas homogen sehingga membutuhkan modifikasi pada metode pengukuran *occupancy*. Kepadatan didefinisikan sebagai seberapa banyak kendaraan di jalan raya dengan ukuran panjang unit dan tanpa mempertimbangkan karakteristik lalu lintas heterogen. Modifikasi pengukuran *area occupancy* dilakukan dengan mempertimbangkan waktu kendaraan melintasi *detector*, ukuran kendaraan dan luas jalan raya sehingga dapat mewakili perilaku lalu lintas dengan cara yang lebih realistis. Perhitungan *area occupancy* dapat digunakan sebagai ukuran untuk mengubah perhitungan lalu lintas homogen ke lalu lintas heterogen namun masih membutuhkan validasi lebih lanjut (Malikarjuna dan Rao, 2006).

Pengukuran *area occupancy* lalu lintas menggunakan teknik simulasi yang dikembangkan untuk mengetahui ruang jalan yang dipergunakan kendaraan pada lalu lintas, pengukurannya menggunakan pertimbangan kecepatan dan memperhitungkan dimensi kendaraan. Metode pengukuran *area occupancy* dilengkapi dengan simulasi sehingga hasilnya dapat dipergunakan untuk mendeteksi lalu lintas tanpa mempertimbangkan lajur lalu lintas (Arasan dan Dhivya, 2009).

2.1.6. Penerapan Metode SVR

Metode SVR digunakan pada model prediksi kepadatan lalu lintas mengintegrasikan antara metode SVR dan *genetic algorithm-simulated anealing*

(GASA). Metode SVR memiliki keunggulan dapat mereduksi nilai eror pada algoritma regresi tunggal, namun untuk meningkatkan akurasi metode SVR dipadukan dengan metode GA-SA bertujuan untuk meraih performa yang lebih baik dan memiliki parameter yang akurat, salah satu keunggulan algoritma SVRGA-SA ini adalah performa dan kemampuan yang meningkat dalam pencarian nilai. Metode SVRGASA dalam penerapannya masih memiliki kekurangan diantaranya belum menggunakan parameter yang akurat dan memuaskan seperti batasan kecepatan berkendara, persentase kendaraan berat melintas dan pengaruh waktu tunggu lampu lalu lintas (Hong, 2011).

Metode SVR dapat dipergunakan untuk mendeteksi *area occupancy* pada *smart building*, dengan menerapkan beberapa set parameter yang berbeda dari data yang dihasilkan pada simulasi perangkat lunak EnergyPlus. Data simulasi hasil pengolahan perangkat lunak EnergyPlus digunakan untuk memproduksi pelatihan data dalam menentukan *area occupancy* diantaranya *solar factor*, *light factor*, *outdoor temperature*, *people occupancy*, *lectrical factor*, *infiltration*, *infrared* dan *HVAC power*. Sedangkan data seleksi parameternya adalah *solar factor*, *light factor*, *outdoor temperature*, *people occupancy*. SVR membantu mendeteksi *occupancy* pada *smart building* dengan melakukan pengaturan parameter tanpa membutuhkan data set presisi tinggi dari sensor *EnergyPlus* sehingga hasilnya perhitungannya memiliki performa tinggi (Hua dkk., 2016).

2.2. Dasar Teori

2.2.1. *Intelligent Transportation System* (ITS)

Intelligent Transportation System (ITS) termasuk sebagai induk dalam upaya untuk menyelesaikan permasalahan lalu lintas, ITS berperan untuk meningkatkan keamanan dan kenyamanan pengguna kendaraan, mereduksi waktu dalam transportasi dan mereduksi biaya. ITS dapat menyajikan informasi secara menyeluruh dan memecahkan permasalahan di jalan raya. ITS merupakan sebuah sistem yang terdiri dari sub sistem pendukung diantaranya adalah:

1. *Advance Traveller Information System* (ATIS) merupakan sebuah sistem yang dapat menganalisis dan memperoleh informasi yang ditujukan bagi pengemudi antara lain informasi kemacetan lalu lintas, kecelakaan, rute, jadwal, navigasi dan informasi *realtime*.

2. *Advance Transport Management System (ATMS)* merupakan sistem manajemen transportasi canggih yang mampu mengontrol perangkat lalu lintas di jalan raya. sistem tersebut bekerja berdasarkan informasi dari sensor dan kamera secara *realtime* untuk pengelolaan lalu lintas. ATMS meliputi pelanggaran lalu lintas, menghitung kendaraan melintas, klasifikasi kendaraan, *adaptif traffic Light* . *Data streaming* lalu lintas *realtime* terkontrol dalam *Transport Management Center (TMC)* yang memungkinkan *stakeholder* dan pengguna jalan raya secara *realtime* menerima *update* tentang kondisi lalu lintas di jalan raya.
3. *Enabled Transport Pricing System (ETPS)* merupakan sistem yang berperan dalam teknologi pembiayaan pada transportasi diantaranya adalah sistem pembayaran otomatis pada gerbang tol dan penggunaan kartu *cashless*.
4. *Advance Public Transportation System (APTS)* merupakan sistem yang mengintegrasikan Moda transportasi dengan teknologi informasi sehingga mampu melaporkan informasi posisi atau jadwal kedatangan dan keberangkatan kepada pengguna transportasi dan pengelola transportasi.

ITS secara umum memproduksi data dalam jumlah besar yang di namakan *big data*, data tersebut akan memiliki dampak signifikan terhadap desain dan penerapan transportasi cerdas. Sub sistem pendukung ITS memberikan gambaran umum terhadap permasalahan apa saja yang dapat diselesaikan oleh teknologi ITS yaitu meningkatkan lalu lintas aman, efisien dan menguntungkan. Dengan bantuan data dan analitik berguna untuk menganalisis kecelakaan, memperhitungkan kepadatan lalu lintas, perencanaan transportasi publik dan perencanaan pada rute bepergian.

2.2.2. *Advance Traffic Management System (ATMS)*

ATMS merupakan sub bidang dalam wilayah ITS, ATMS juga berperan dalam *Intellegent Vehicles-Highway System (IVHS)* untuk efisiensi lalu lintas. ATMS merupakan teknologi terintegrasi dalam mengoptimalkan lalu lintas, mendeteksi kepadatan lalu lintas, mengusulkan rute terbaik dan mengurangi tingkat kecelakaan agar tidak menghambat mobilitas. ATMS dibagi menjadi empat bagian yaitu pengontrol lalu lintas, informasi pengemudi, manajemen insiden dan koordinasi antar lembaga yang berkaitan detail informasinya sebagai berikut:

1. Pengontrol lalu lintas

Pengontrol lalu lintas terdiri dari: Pusat pengawasan atau sistem deteksi, lalu lintas sinyal kontrol terpusat, pantauan CCTV, pengarah rute (*guidance system*), *Decision Support System (DSS)*, *expert system* dan strategi.

2. Informasi pengendara

Informasi pengendara terdiri dari: *variable message sign (VMS)*, radio informasi jala raya, sistem informasi pengendara dan sistem navigasi pengendara.

3. Manajemen insiden

Manajemen insiden terdiri dari: deteksi insiden otomatis, perencanaan manajemen insiden layanan patroli, *DSS dan expert system*.

4. Koordinasi antar lembaga yang berkaitan

Bentuk koordinasi terdiri dari: pemusatan sistem informasi, *sharing resources*, *DSS dan expert system*.

ATMS wajib dioperasikan dan dipertahankan apabila manfaat yang diraih terbukti (Kay, 1990). Berbagai teknologi dalam ATMS telah dikembangkan dan dibekali kemampuan kognitif sehingga mampu menganalisis dan memutuskan secara otomatis. Peningkatan efisiensi ATMS dapat dilakukan dengan cara mengintegrasikan teknologi ATMS pada area lokal sehingga memudahkan untuk terjadinya koordinasi dan mendapatkan informasi secara cepat. Umumnya peningkatan tersebut dapat mengurangi tingkat kecelakaan terkait kendaraan bermotor di jalan raya dan peningkatan efisiensi operasional unit-unit tanggap darurat.

2.2.3. Kategori Kendaraan

Dokumen Manual Kapasitas Jalan Indonesia (MKJI) (Direktorat Jenderal Bina Marga., 1997) merupakan panduan yang diterapkan sebagai sarana dalam perancangan, perencanaan dan analisa fasilitas lalu lintas. Dokumen tersebut menjelaskan bahwa kendaraan merupakan sebuah unsur pada lalu lintas, setiap kendaraan memiliki pengemudi yang menentukan dinamika lalu lintas karena kendaraan yang dikemudikan memiliki karakteristik kecepatan, percepatan, perlambatan, dimensi dan beban. Kendaraan dibagi menjadi beberapa jenis

diantaranya adalah kendaraan ringan, kendaraan berat, sepeda motor dan kendaraan tak bermotor. Berikut kategori kendaraan yang ditunjukkan pada Tabel 2.1.

Tabel 2.1. Kategori kendaraan

Kategori Kendaraan	Kendaraan	Keterangan
LV	<i>Light Vehicle</i> (Kendaraan ringan)	Kendaraan dengan ciri 4 roda memiliki 2 as dengan jarak as roda 2 – 3 meter, diantaranya mobil penumpang, pickup, truk kecil sesuai dengan klasifikasi bina marga.
HV	<i>Heavy Vehicle</i> (Kendaraan berat)	Kendaraan lebih dari 4 roda, diantaranya truk 2 as, truk 3 as, truk kombinasi sesuai klasifikasi bina marga.
MC	<i>Motor Cycle</i> (Sepeda motor)	Diantaranya sepeda motor dan kendaraan roda 3, sesuai klasifikasi bina marga.
UM	<i>Unmotorize</i> (Kendaraan tak bermotor)	Jenis kendaraan yang gerakan oleh orang maupun hewan, antara lain sepeda, becak, kereta kuda dan kereta dorong sesuai klasifikasi bina marga.

Selain kategori kendaraan yang telah ditetapkan oleh MKJI, berdasarkan Keputusan menteri PU No. 370/KPTS/M/2007 juga menggolongkan kendaraan yang biasa diterapkan untuk identifikasi tarif pembayaran di jalan tol dan tarif angkutan penyeberangan. Relasi data pada Tabel 2.1 dan Tabel 2.2 adalah menerjemahkan tabel jenis kendaraan untuk mempermudah mengetahui dimensi kendaraan yang beragam. Penelitian ini memerlukan data dimensi kendaraan sebagai cara untuk menghitung area hunian (*Occupancy*) di jalan raya (Transport for NSW, 2012). Data dimensi kendaraan ditunjukkan pada Tabel 2.2.

Tabel 2.2. Dimensi kendaraan

Kategori kendaraan	Keterangan	Panjang dan lebar (m)	Luas <i>Occupancy</i> (m ²)
LV1	Sedan, Pick up/truk kecil dan bus	4,7 x 1,7	7,99
LV2	Truk dengan 2 (dua) gandar	5,8 x 2,1	12,41
HV1	Truk dengan 3 (tiga) gandar	12 x 2,5	30,00
HV2	Truk dengan 4 (empat) gandar	12,0 x 2,6	31,20
HV3	Truk dengan 5 (lima) gandar	21 x 2,6	54,00
MC	Kendaraan bermotor roda 2 (dua)	1,75 x 0,7	1,22

2.2.4. Satuan Mobil Penumpang (SMP)

Kendaraan merupakan faktor penentu terhadap dampak lalu lintas sehingga untuk mengetahui arus lalu lintas maka memerlukan perhitungan berdasarkan satuan mobil penumpang yaitu penyetaraan satuan kendaraan dalam arus lalu lintas ke kendaraan ringan (mobil penumpang) yang dihitung berdasarkan faktor pengali berbagai jenis kendaraan menggunakan nilai yang telah ditentukan oleh EMP. Nilai pengali dipengaruhi oleh tipe atau jenis kendaraan dan dimensi kendaraan. Salah satu kegunaan dari smp adalah untuk menentukan volume kapasitas suatu ruas jalan.

2.2.5. Ekuivalensi Mobil Penumpang (EMP)

Merupakan teknik untuk mengkonversi berbagai tipe kendaraan untuk diseragamkan menjadi tipe kendaraan ringan dengan menggunakan nilai koefisien tentunya dengan mempertimbangkan faktor-faktor yang mempengaruhinya seperti jalur dua arah dan lebar jalan, nilai emp ditunjukkan pada Tabel 2.3.

Tabel 2.3. Ekuivalensi Mobil Penumpang (EMP)

Tipe Jalan	Arus Lalu lintas total dua arah (kend/jam)	Emp			
		Kendaraan ringan (LV)	Kendaraan berat (HV)	Sepeda motor (MC)	
				Lebar jalur (m)	
≤ 6	> 6				
Dua lajur dua arah (2/2 D)	0 s.d 1800	1,0	1,3	0,5	0,4
	≥ 1800	1,0	1,2	0,35	0,25

2.2.6. Karakteristik Jalan

Jalan adalah lintasan yang dilalui oleh kendaraan bermotor maupun tidak bermotor termasuk pejalan kaki, fungsinya adalah mengalirkan arus lalu lintas secara lancar dan sanggup menopang beban muatan pada kendaraan, sedangkan kapasitas jalan merupakan kekuatan ruas jalan untuk menampung volume lalu lintas dalam satu waktu tertentu, jalan yang dipergunakan pada penelitian ini adalah jalan perkotaan.

Jalan perkotaan dibagi menjadi 2 (dua) segmen diantaranya segmen jalan perkotaan atau semi perkotaan dan segmen jalan luar kota. Masing masing segmen memiliki ciri tersendiri yaitu :

1. Segmen jalan perkotaan atau semi perkotaan

Jalan perkotaan mempunyai ciri-ciri khusus yaitu perkembangannya secara permanen baik dari seluruh dan sepanjang jalan atau sebagian kecil pada jalan. Jalan tersebut terletak di dekat pusat kota atau di daerah perkotaan dengan ukuran mencapai 100.000 penduduk.

2. Segemen jalan luar kota

Memiliki ciri tidak ada perkembangan yang terus menerus pada bagian sisi jalan. Karakteristik lalu lintas pada jalan kota atau semi kota adalah terjadinya arus lalu lintas yang padat antara pagi hari dan sore hari. Dapat diartikan terdapat perubahan secara umum terhadap komposisi lalu lintas (lebih tinggi kendaraan pribadi dibandingkan kendaran berat).

2.2.6.1 Jalan Dua Lajur Dua Arah (2/2 Terbagi)

Merupakan jalan yang berada di perkotaan memiliki karakteristik dua jalur dan dua arah dengan lebar jalan antara 7 - 10.5 meter. Tipe jalan dapat menunjukkan kinerja berbeda dalam menentukan beban pada lalu lintas tertentu. Untuk melakukan perhitungan pada jalan dua jalur dan dua arah ini membutuhkan prosedur pemilihan perhitungan jalan, yaitu untuk jalan terbagi akan dilakukan perhitungan secara terpisah pada masing-masing arah lalu lintas untuk menghitung volume yang menunjukkan jumlah kendaraan yang melewati suatu titik pengamatan dalam satuan waktu. Rincian data tipe jalan ditunjukkan pada Tabel 2.4.

Tabel 2.4. Jalan dua jalur dan dua arah (2/2 D)

Kondisi Jalan	Keterangan
Lebar jalur lalu lintas	7 meter
Lebar bahu efektif	Paling sedikit 2 meter pada setiap sisi
Pemisah arah	50 - 50
Hambatan samping	Rendah
Ukuran kota	1,0 – 3,0 Juta
Tipe alinyemen	Datar

2.2.7. Volume Lalu Lintas

Volume lalu lintas adalah jumlah kendaraan yang melawati titik atau jalan raya dalam kurun waktu tertentu, volume lalu lintas dapat dinyatakan dalam periode tahunan, harian, jam dan menit. Sehingga dapat digunakan untuk mengetahui lalu lintas harian rata-rata sebagai dasar untuk analisa perencanaan jalan.

2.2.8. Arus Lalu Lintas

Secara umum arus lalu lintas adalah rata-rata kendaraan melewati titik tertentu di jalan dengan interval pengambilan data kurang dari 1 jam atau biasanya 15 menit. Arus lalu lintas memiliki karakter tidak seragam akan tetapi berbeda dalam ruang dan waktu, karena itu pengambilan sampel dilakukan secara acak. Pada kenyataannya, karakteristik lalu lintas tersirat sebagai arus, kecepatan dan volume atau konsentrasi kendaraan sebagai parameter dari distribusi statistik yang tidak memiliki angka mutlak. Arus lalu lintas ini dikumpulkan langsung melalui titik yang memerlukan pengukuran dari waktu ke waktu. Kesulitan utama dalam menerapkan pendekatan ini adalah menentukan lokasi secara akurat, meskipun hanya dapat memberikan informasi kecepatan kendaraan secara realistis, secara umum tidak mungkin untuk satu kendaraan dapat teridentifikasi laju kecepatannya atau kepadatannya secara pasti. Akan tetapi dengan menggunakan sensor yang tepat, setiap kendaraan dapat terekam pada waktu dan jarak antar kendaraannya sendiri. Dengan demikian masih ada masalah dalam menemukan informasi perbandingan arus atau kepadatan (Hall 1992).

2.2.9. Mengukur Arus Lalu Lintas

Arus lalu lintas secara umum dipengaruhi oleh berbagai macam variasi kendaraan seperti kendaraan berat yang memiliki ukuran yang besar dan kecepatan lambat, untuk menghitung akurasi arus lalu lintas perlu mempertimbangkan pengaruh kendaran berat, variasi lalu lintas dan karakteristik pengemudi. Pengukuran arus lalu lintas biasanya menggunakan satuan jumlah kendaraan per jam dan semua kendaraan yang melintas disetarakan dengan mobil penumpang. Pengukuran arus lalu lintas dapat dilihat pada Persamaan 2.2.

$$V_p = \frac{V}{PHF * N * f_{HV} * f_p} \quad (2.2)$$

Diketahui bahwa V_p merupakan arus lalu lintas (smp/jam/lajur), V merupakan volume per jam, PHF adalah faktor jam puncak atau faktor waktu puncak kendaraan (*peak hour factor*), N jumlah lajur, f_{HV} penyesuaian kendaraan berat dan f_p adalah pengaruh faktor pengemudi.

2.2.10. Faktor Jam Puncak (PHF)

Merupakan representasi dari *peak hour factor* (PHF) yaitu konversi pengamatan volume lalu lintas selama 1 jam yang dipecah menjadi pengamatan 15 menit berurutan, nilai PHF adalah nilai tertinggi arus lalu lintas selama 4x waktu pengamatan 15 menit (Iskandar, 2015). Pada jalaur landai nilai PHF berkisar antara 0,80 hingga 0,95, PHF dapat dikehui menggunakan persamaan 2.3.

$$PHF = \frac{V}{[4 * v_{15}]} \quad (2.3)$$

Diketahui PHF faktor jam punyak volume lalu lintas, V total volume lalu lintas selama 1 jam, v_{15} adalah volume tertinggi selama 4 x 15 menit.

2.2.11. Penyesuaian Kendaraan Berat (f_{HV})

Lalu lintas heterogen memiliki ciri-ciri kendaran yang melintas berbeda-beda tipe kendaraan untuk memperoleh akurasi dalam perhitungan arus lalu linas maka diperukan penyetraan kendaraan menggunakan unit mobil penumpang per jalur per jam. Penyetaraan diperoleh menggunakan Persamaan 2.4.

$$f_{HV} = \frac{1}{1 + P_T (E_T - 1) + P_R (E_R - 1)} \quad (2.4)$$

Diketahui bahwa f_{HV} adalah nilai penyesuaian kendaraan berat, E_T dan E_R merupakan equivalensi mobil penumpang untuk truk, bus dan kendaraan rekreasi, P dan P_R merupakan proporsi kendaraan truk, bus atau kendaraan rekreasi dalam arus lalu lintas. dalam beberapa kasus untuk menentukan equivalensi juga mempertimbangkan kondisi jalan diantaranya medan landai, medan berliku dan

medan pegunungan, masing-masing memiliki equivalensi yang ditunjukkan pada Tabel 2.5.

Tabel 2.5. EMP mengacu pada spesifik medan

Faktor	Tipe medan		
	Landai	Berliku	Medan pegunungan
E_T truk dan bus	1,5	2,5	4,5
E_R Kendaraan rekreasi	1,2	2,0	4,0

2.2.12. Faktor Populasi Pengemudi (f_p)

Pada prinsipnya faktor pengemudi berdampak langsung terhadap kapasitas di jalan raya sehingga mempengaruhi perhitungan arus lalu lintas, hal ini terjadi ketika pengemudi memilih menyesuaikan kendaraannya dan beradaptasi ketika bertemu dengan pengemudi lain. Oleh sebab itu perlunya faktor penyesuaian yang digunakan untuk mencerminkan efek tersebut. Secara umum rentang nilai f_p dari 0,85 hingga 1,00. Nilai yang biasa dipilih oleh analis adalah 1,00 kecuali ada kondisi khusus dimana keadaan jalan mempengaruhi pengemudi (Hcm, 2000) (Heaslip dkk., 2008).

2.2.13. Menentukan kecepatan arus bebas (FFS)

Kecepatan rata-rata kendaraan dilakukan dengan mengukur kecepatan kendaraan pada arus rendah hingga arus sedang selama waktu yang dibutuhkan pada segmen jalur bebas hambatan. Pengukuran kecepatan meliputi semua jalur dan wajib memperhatikan penyesuaian kondisi jalan seperti lebar lajur, jumlah jalur, kerapatan lalu lintas serta hal lain yang dapat mempengaruhi kecepatan sebagai contoh kendaraan berat dapat mempengaruhi kecepatan kendaraan lain yang melintas.

Pengukuran kecepatan dapat dilakukan dengan dua acara yaitu pengukuran di lapangan secara langsung dan pengukuran berdasarkan aturan yang disediakan oleh manual kapasitas jalan raya. Apabila pengukuran lapangan tidak mungkin dilakukan maka untuk menentukan rata-rata kecepatan dapat diperkirakan dengan mempertimbangkan karakter fisik jalan raya seperti lebar lajur, hambatan, jumlah

lajur, penyesuaian lebar bahu jalan dan kerapatannya. Perhitungan kecepatan rata-rata menggunakan Persamaan 2.5.

$$FFS = BFSS - F_{LW} - F_{LC} - F_N - F_{ID} \quad (2.5)$$

Diketahui bahwa FFS adalah kecepatan arus bebas, $BFSS$ nilai penyesuaian kecepatan arus dasar yaitu 110 km/jam pada jalur perkotaan, F_{LW} nilai penyesuaian lebar lajur, F_{LC} nilai penyesuaian lebar bahu kiri jalan, F_N nilai penyesuaian jumlah lajur dan F_{ID} nilai jarak spasi antara kendaraan satu dengan kendaraan lainnya (*interchange density*) (Hcm, 2000).

2.2.14. Penyesuaian Lebar Lajur (F_{LW})

Pada umumnya lebar lajur di jalan raya berukuran 3,6 meter. Apabila nilai kecepatan dasar $BFSS$ yang ditentukan pada jalur perkotaan adalah 110 Km/jam dan lebar lajur 3,6 meter maka nilai reduksi kecepatannya adalah 0,0. Jika lebar lajur kurang dari 3,6 meter maka nilai koefisien reduksi kecepatan dapat ditunjukkan oleh Tabel 2.6.

Tabel 2.6. Penyesuaian kecepatan terhadap lebar lajur

Panjang Lajur (meter)	Nilai Reduksi Kecepatan F_{LW} km/jam
3,6	0,0
3,5	1,0
3,4	2,1
3,3	3,1
3,2	5,6
3,1	8,0
3,0	10,6

2.2.15. Penyesuaian Lebar Bahu Kiri Jalan (F_{LC})

Lebar bahu jalan pada umumnya memiliki ukuran 1,8 meter, apabila lebar bahu jalur kiri lebih kecil 1,8 meter maka berpengaruh pada nilai kecepatan dasar $BFSS$. Tidak ada aturan penyesuaian yang tersedia untuk mencerminkan efek jarak median lateral jalan kurang dari 0,6 m. Jarak bebas lateral kurang dari 0,6 m disisi

kanan atau kiri jalan bebas hambatan dianggap jarang. Pertimbangan yang cukup harus digunakan dalam menentukan apakah objek atau penghalang di sepanjang sisi kanan jalan bebas hambatan merupakan penghalang atau yang sebenarnya. Penghalang tersebut seperti dinding penahan, penghalang beton, atau pagar pembatas, atau mungkin tidak kontinu, seperti penyangga ringan atau penyangga jembatan. Dalam beberapa kasus, pengemudi mungkin menjadi terbiasa dengan jenis penghalang tertentu, dalam hal ini pengaruhnya terhadap arus lalu lintas dapat diabaikan. Nilai penyesuaian kecepatan ditentukan pada Tabel 2.7.

Tabel 2.7. Nilai penyesuaian kecepatan terhadap lebar bahu jalan

Lebar Bahu Jalur Kiri (meter)	Nilai Reduksi Kecepatan F_{LC} km/jam			
	Jalur Satu Arah			
	2	3	4	≥ 5
$\geq 1,8$	0,0	0,0	0,0	0,0
1,5	1,0	0,7	0,3	0,2
1,2	1,9	1,3	0,7	0,4
0,9	2,9	1,9	1,0	0,6
0,6	3,9	2,6	1,2	0,8
0,3	4,8	3,2	1,6	1,1
0,0	5,8	3,9	1,9	1,3

2.2.16. Penyesuaian Jumlah Lajur Dalam Satu Arah (F_N)

Memperhitungkan jumlah lajur jalan raya menjadi penting karena berpengaruh pada nilai kecepatan dasar $BFFS$, jumlah lajur yang ditentukan adalah lajur utama, lajur pelengkap atau tambahan tidak dipertimbangkan.. Nilai penyesuaian $BFFS$ ditunjukkan oleh Tabel 2.8.

Tabel 2.8. Tabel penyesuaian jumlah lajur

Jumlah Lajur	Nilai Reduksi Kecepatan F_N km/jam
≥ 5	0,0
4	2,4
3	4,8
2	7,3

Penyesuaian nilai FFS pada Tabel 2.8. hanya diperlukan pada jalan tipe bebas hambatan pada perkotaan, untuk memperkirakan pada jalan bebas hambatan pedesaan maka nilai penyesuaian jumlah lajur pada F_N menjadi 0,0.

2.2.17. Penyesuaian Perubahan Kepadatan Lalu Lintas (F_{ID})

Kecepatan arus bebas di jalan raya bisa berkurang tergantung pada kondisi lalu lintas, hal ini dipengaruhi oleh kepadatan lalu lintas perlahan meningkat. Nilai dasar perubahan kepadatan lalu lintas ditunjukkan oleh Tabel 2.9.

Tabel 2.9. Tabel penyesuaian perubahan kepadatan lalu lintas

Perubahan Per Kilometer	Nilai Reduksi Kecepatan F_{ID} km/jam
$\leq 0,3$	0,0
0,4	1,1
0,5	2,1
0,6	3,9
0,7	5,0
0,8	6,0
0,9	8,1
1,0	9,2
1,1	10,2
1,2	12,1

2.2.18. Menentukan Tingkat Pelayanan Jalan

Langkah dalam menentukan tingkat pelayanan jalan atau LOS dapat diketahui dengan melakukan perbandingan arus lalu lintas dengan kecepatan rata-rata. Perhitungan arus lalu lintas menggunakan Persamaan 2.6.

$$D = \frac{V_p}{S} \quad (2.6)$$

Diketahui bahwa D adalah nilai kepadatan satuan mobil penumpang per jam per lajur, V_p merupakan nilai arus lalu lintas satuan mobil penumpang per jam per lajur dan S rata-rata kecepatan kendaraan per kilometer per jam.

2.2.19. Pengukuran Lalu Lintas *Homogen* Dan *Heterogen*

Manajemen lalu lintas yang sudah digunakan di negara maju namun studi tersebut terbatas karena umumnya di negara maju lalu lintas cenderung homogen, sedangkan kondisi lalu lintas sebenarnya sangat bervariasi atau heterogen dan menyebabkan karakteristik mikroskopis seperti waktu tempuh kendaraan dan jarak antar kendaraan sulit diukur, sekalipun menggunakan karakteristik lalu lintas makroskopik seperti kepadatan ataupun *occupancy*. Oleh sebab itu diperlukan pengukuran yang tepat untuk mewakili sifat lalu lintas yang heterogen diberbagai dimensi kendaraan serta tidak mematuhi disiplin jalur. Dengan demikian diusulkan *area occupancy* sebagai pengukuran yang dimodifikasi untuk mempertimbangkan berbagai dimensi lalu lintas kendaraan. Berdasarkan pengamatan di lapangan menunjukkan bahwa *area occupancy* lebih berarti dalam merepresentasikan lalu lintas *heterogen* dibandingkan *occupancy*.

2.2.20. Menghitung *Occupancy*

Occupancy didefinisikan sebagai persentase waktu ruas jalan yang ditempati oleh kendaraan selama periode waktu tertentu. *Occupancy* setara dengan kepadatan dengan syarat kondisinya seimbang (*ekuilibrium*), namun apabila kondisinya tidak seimbang maka relasi atau hubungan antara *occupancy* dan kepadatan relatif hanya sedikit. Hal ini dapat dilihat melalui persamaan 2.7.

$$\rho = \frac{\sum_{i=1}^N O_i}{T} \tag{2.7}$$
$$O_i = \frac{l_i + d}{v_i}$$

Diketahui bahwa ρ *occupancy*, O_i waktu kendaraan ke i menduduki detector (satuan waktu), l_i panjang kendaraan ke i , d Panjang detector (meter), v_i kecepatan kendaraan ke i , T adalah waktu pengamatan. Ketika kepadatan dan tingkat *occupancy* diukur pada wilayah pengukuran umum (per waktu dan jarak) (Malikarjuna dan Rao, 2006).

2.2.21. Menghitung *Area occupancy*

Konsep *area occupancy* dapat diterapkan untuk menghitung kepadatan pada kondisi lalu lintas heterogen. *Area occupancy* lalu lintas heterogen dapat diperkirakan menggunakan simulasi arus lalu lintas untuk berbagai tingkat volume dengan komposisi lalu lintas representatif yang berlaku di jalan perkotaan.

Area occupancy dapat digunakan untuk mengetahui karakteristik lalu lintas, dengan menggunakan relasi perhitungan ruang (penempatan kendaraan di jalan raya) dan waktu (waktu kendaraan melintas), *area occupancy* dapat dihitung menggunakan persamaan 2.8. (Malikarjuna dan Rao, 2006).

$$\rho A^{(A)} = \frac{\sum_{i=1}^N \frac{L - x_i}{\bar{v}_i} * w_i * l_i}{T * W * L} \quad (2.8)$$

Diketahui bahwa $\rho A^{(A)}$ adalah *area occupancy* yang diukur berdasarkan ruang dan waktu, L panjang jalan, W lebar jalan raya, x_i adalah sisa jarak yang ditempuh kendaraan selama melintasi *detektor*, Jika waktu pengamatan ditentukan selama 1 menit dan kecepatan kendaraan 90 Km/jam maka nilai x_i adalah 0 (nol), karena dapat dipastikan kendaraan tersebut dapat melintasi *detektor* dalam waktu kurang dari 1 menit. Sebaliknya kendaraan dengan kecepatan rendah tidak dapat melintasi area *detektor* dalam waktu 1 menit maka nilai x_i adalah sisa jarak yang harus ditempuh. \bar{v}_i kecepatan seluruh kendaraan ketika melintasi area *detektor* (*space mean speed*). w_i, l_i Panjang dan lebar kendaraan dan T waktu pengamatan. Dengan demikian *area occupancy* dapat dihitung persentasenya dan disetarakan dengan perhitungan arus kendaraan sehingga dapat merujuk pada tingkat pelayanan jalan.

2.2.22. Tingkat Pelayanan Jalan (*Level Of Service*)

Tingkat pelayanan jalan merupakan cara yang digunakan untuk mengukur kinerja jalan. Ukuran terbagi menjadi 6 (enam) level tingkat pelayanan mulai dari A, B, C, D dan E (HCM, 2000). Kelas A mewakili arus lalu lintas lancar dan tingkat E mewakili kepadatan atau kemacetan. Data tingkat pelayanan jalan ditunjukkan pada Tabel 2.10.

Tabel 2.10. Tabel tingkat pelayanan jalan raya

Kriteria	LOS				
	A	B	C	D	E
Kecepatan arus bebas = 120 Km/jam					
Maksimum kepadatan (Kendaraan/Km)	7	11	16	22	28
Kecepatan minimum (Km/jam)	120.0	120.0	114.6	99.6	85.7
Maksimum volume kapasitas (V/C)	0.35	0.55	0.77	0.92	1.00
Maksimum arus (Kendaraan /jam)	840	1320	1840	2200	2400
Kecepatan arus bebas = 110 Km/jam					
Maksimum kepadatan (Kendaraan /Km)	7	11	16	22	28
Kecepatan minimum (Km/jam)	110.0	110.0	108.5	97.2	83.9
Maksimum volume kapasitas (V/C)	0.33	0.51	0.74	0.91	1.00
Maksimum arus (Kendaraan /jam)	770	1210	1740	2135	2350
Kecepatan arus bebas = 100 Km/jam					
Maksimum kepadatan (Kendaraan /Km/lajur)	7	11	16	22	28
Kecepatan minimum (Km/jam)	100.0	100.0	100.0	93.8	82.1
Maksimum volume kapasitas (V/C)	0.30	0.48	0.70	0.90	1.00
Maksimum arus (Kendaraan /jam)	700	1100	1600	2065	2300
Kecepatan arus bebas = 90 Km/jam					
Maksimum kepadatan (Kendaraan /Km/jalur)	7	11	16	22	28
Kecepatan minimum (Km/jam)	90.0	90.0	90.0	89.1	80.4
Maksimum volume kapasitas (V/C)	0.28	0.44	0.64	0.87	1.00
Maksimum arus (Kendaraan /jam)	630	990	1440	1955	2250

Akan tetapi tingkatan pelayanan tidak dapat digunakan pada semua kondisi jalan, sehingga memerlukan standar pengukuran secara seragam berdasarkan *speed mean space*. Penelitian ini menggunakan ukuran tingkat pelayanan jalan sebagai referensi dalam menentukan tingkat kepadatan lalu lintas.

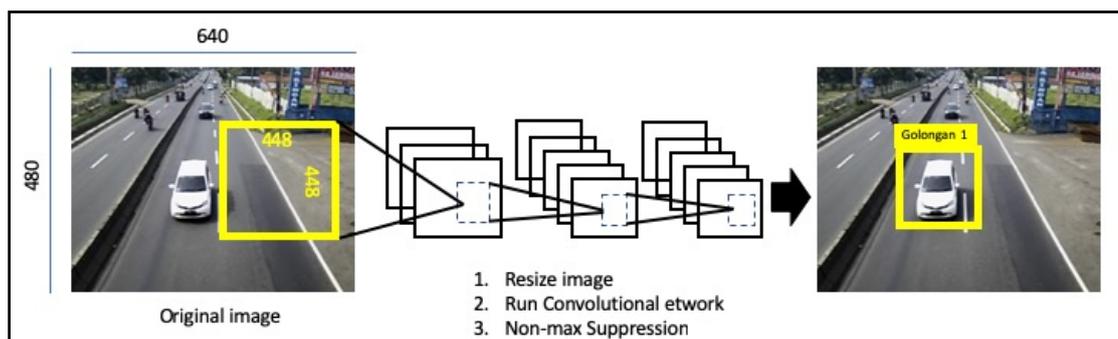
2.2.23. You Only Look Once (YOLO)

YOLO menggunakan jaringan syaraf tiruan tunggal untuk mendeteksi (objek diberi tanda kotak) dan memberikan nilai probabilitas kelas dari sebuah gambar dalam satu kali evaluasi. Keunggulan dari YOLO adalah memiliki kecepatan dalam

mendeteksi obyek dan mampu mengolah video *realtime* sebanyak 45 *frames per second* (FPS) pada jaringan kompleks. Dalam jaringan yang lebih kecil YOLO mampu memperoleh 155 FPS apabila dibandingkan rata-rata pendeteksian secara *realtime* pada metode lain. Kekurangan YOLO adalah kecepatan dalam pemrosesan saat mendeteksi obyek berakibat pada kemungkinan terjadinya *error* dan permasalahan akurasi. Namun YOLO masih mengungguli metode deteksi lainnya seperti *deformable parts model* (DPM) dan *regional based convolutional neural network* (R-CNN) (Redmon dkk., 2015). YOLO9000 atau YOLOv2 mengalami berbagai perbaikan pada metode deteksinya, pada sistem deteksi *realtime* YOLOv2 mampu mendeteksi lebih dari 9000 obyek. YOLOv2 menerapkan pelatihan multi skala yang dapat bekerja pada berbagai ukuran gambar sehingga menawarkan kecepatan dan akurasi (Redmon dan Farhadi, 2017). YOLOv3 muncul dengan beberapa pembaharuan. Banyak perubahan desain untuk membuat lebih baik, melatih jaringan baru yang lebih besar daripada YOLO versi sebelumnya sehingga lebih akurat dan cepat (Redmon dan Farhadi, 2018).

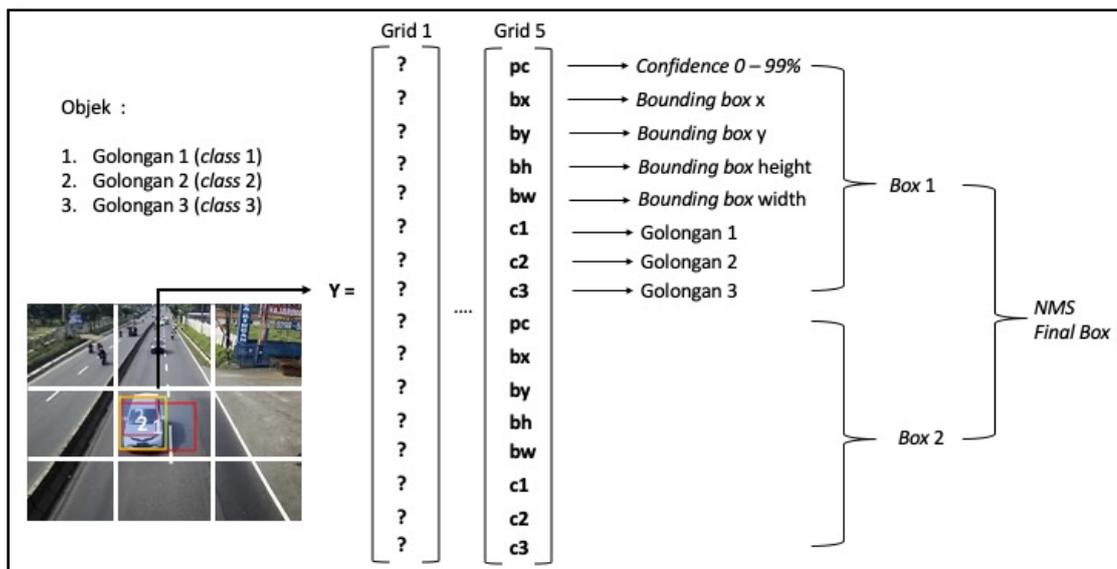
2.2.23.1 Cara Kerja YOLO

Pengolahan gambar pada metode YOLO meliputi tiga hal, Pertama YOLO merubah ukuran gambar menjadi 448 x 448 piksel, Kedua menjalankan jaringan syaraf tunggal dan, ketiga adalah menyaring *bounding box* dengan mengaplikasikan *non maximum suppression* (NMS) dengan menerapkan nilai *confidence predict* lebih dari 0.6 atau 60% pada model YOLO sehingga hanya *bounding box* terpilih yang akan muncul. Selanjutnya menyingkirkan *bounding box* melalui proses *intersect over union* (IOU) dengan nilai *threshold* melebihi 0,5. Pada kondisi ini secara otomatis *bounding box* sesuai dengan *ground truth*. Gambar 2.1. menunjukkan sistem deteksi YOLO.



Gambar 2.1. Deteksi YOLO (Redmon dkk., 2015)

YOLO secara otomatis membagi sebuah *input* gambar tersebut menjadi $S \times S$ *grid*. Jaringan syaraf tunggal dijalankan secara serentak untuk mendapatkan berbagai prediksi *bounding box* pada *grid cell* tersebut. Setiap *grid cell* terdiri dari 2 (dua) buah prediksi *bounding box* sehingga apabila suatu obyek jatuh pada sebuah *grid*, maka *grid cell* tersebut yang bertanggung jawab terhadap hasil prediksi pada gambar. setiap *bounding box* menghasilkan 5 (lima) nilai prediksi yaitu : x, y, w, h dan nilai *confidence*. Nilai (x, y) mewakili sumbu x dan sumbu y yang merepresentasikan sebuah titik tengah pada *bounding box* dalam *grid cell*. Nilai (w, h) mewakili lebar dan tinggi *bounding box*. Gambar 2.2 merupakan ilustrasi proses *bounding box* pada YOLO.



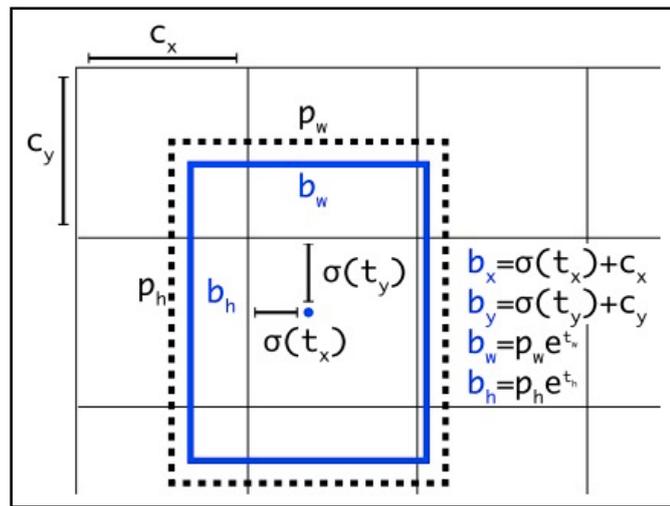
Gambar 2.2. Properti parameter pada metode YOLO

Sehingga apabila memiliki *grid cell* sebanyak 3×3 maka akan menghasilkan persamaan $3 \times 3 \times 2 \times 8$ *tensor* gambar.

Pada saat pengujian, YOLO mengalikan probabilitas kelas (menentukan kelas bersyarat) dan prediksi individu *bounding box*, maka dihasilkan Persamaan 2.9 sebagai berikut.

$$\begin{aligned}
 & Pr(Class_i|Object) * Pr(Object) * IOU \frac{truth}{pred} \\
 & = Pr(Class_i) * IOU \frac{truth}{pred}
 \end{aligned}
 \tag{2.9}$$

Persamaan tersebut menghasilkan skor *confidence* pada tiap-tiap kelas secara spesifik, sehingga *bounding box* yang terprediksi hasilnya cocok dengan obyek, secara formal YOLO memprediksi *confidence* sebagai $Pr(Object) * IOU \frac{truth}{pred}$, apabila tidak terdapat sebuah obyek pada *grid cell* maka skor *confidence* adalah 0 (nol). Begitu pula sebaliknya, apabila terdapat sebuah obyek maka skor *confidence* akan sama dengan *intersection over union* (IOU) pembandingnya diantara prediksi *bounding box* dan *ground truth* ditunjukkan pada Gambar 2.3.



Gambar 2.3. Prediksi lokasi *bounding box* (Redmon dan Farhadi, 2018)

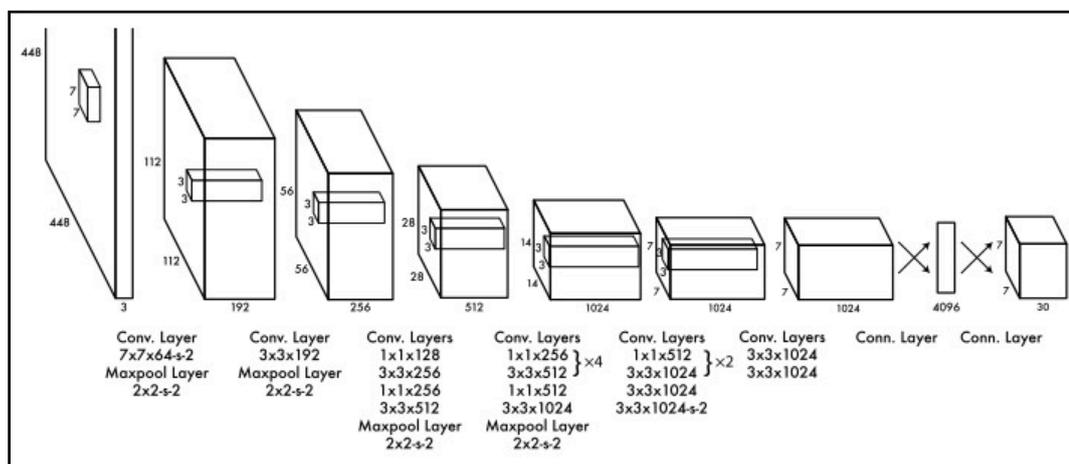
Mengacu pada sistem YOLO versi sebelumnya yaitu sistem prediksi menggunakan dimensi *cluster* sebagai sebuah *anchor box*. *Network* YOLO memprediksi 4 (empat) buah titik koordinat untuk setiap *bounding box* yaitu (t_x , t_y , t_w , t_h). Apabila sebuah *cell* dikatakan *offset* (*bounding box* melebihi batasan dari sebuah *cell*) dari sudut kiri atas gambar diterjemahkan oleh titik (c_x , c_y) dan *bounding box* sebelumnya yang memiliki tinggi dan lebar, maka prediksi YOLO sesuai dengan Persamaan 2.10.

$$\begin{aligned}
 b_x &= \sigma(t_x) + c_x \\
 b_y &= \sigma(t_y) + c_y \\
 b_w &= p_w e^{t_w} \\
 b_h &= p_h e^{t_h}
 \end{aligned}
 \tag{2.10}$$

Selama melakukan tahap *training*, YOLO menggunakan *sum of squared error loss*. Sebuah *ground truth* pada koordinat prediksi dinyatakan sebagai \hat{t}_* maka nilainya sama dengan *ground truth* di kurangi dengan nilai prediksinya, persamaannya adalah $(\hat{t}_* - t_*)$. Ini merupakan sebuah nilai *ground truth* yang hasilnya didapat dengan cara membalikkan persamaan di atas.

2.2.23.2 Perancangan dan pelatihan pada *network* YOLO

YOLO *network* arsitektur serupa dengan *convolutional neural network* (CNN) yang terinspirasi oleh model *GoogleNet* untuk klasifikasi sebuah gambar yang terdiri dari 24 (dua puluh empat) *convolutional layer* dan 4 (empat) *max pooling layer*, kemudian diikuti oleh 2 (dua) *fully connected layer*. YOLO juga menggunakan *filter* 1 x 1 untuk mereduksi layer, kemudian diikuti oleh *filter* 3 x 3 *convolutional layer*, sehingga menghasilkan 7 x 7 x 2 (*bounding box* pada setiap *cell*) x 5 (sumbu x, y, w, h dan *confidence*) + 20 class (daftar obyek) sama dengan 1470 tensor prediksi yang ditunjukkan pada Gambar 2.4.



Gambar 2.4. Arsitektur YOLO (Joseph Redmon, 2015)

Pelatihan data pada YOLO dibagi menjadi 2 (dua) tahap. Pertama sebelum dilakukan pelatihan YOLO mereduksi gambar sebagai parameter *input* menjadi gambar yang memiliki resolusi 224 x 224 *piksel* untuk proses klasifikasi. Selanjutnya menggunakan 20 (dua puluh) *convolutional layer* yang diikuti *average pooling* dan *connected layer*. Kedua, YOLO melatih untuk model deteksi dengan menambahkan 4 (empat) *convolution layer* dan 2 (dua) *fully connected layer*. Untuk meningkatkan akurasi YOLO menaikkan *input* resolusi menjadi 448 x 448 untuk mendapatkan visual yang jelas pada saat mendeteksi.

Dalam proses pelatihan data, metode YOLO menimbulkan permasalahan yang dinamakan *Loss Function* hal ini dapat menyebabkan *confidence loss*, *localization loss* dan *classification loss*.

1. Confidence loss

Merupakan permasalahan pada deteksi YOLO yaitu jaringan menghasilkan kotak prediksi berukuran besar namun tidak terdapat obyek apa pun sehingga secara otomatis kotak prediksi ini mendapatkan nilai 0 (nol). Untuk mengatasi permasalahan tersebut. Metode YOLO menurunkan nilai *loss* dengan cara melakukan pengaturan pada parameter $\lambda_{noobj} = 0.5$, ditunjukkan pada model persamaan 2.11 yaitu reduksi *Confidence Loss Function*.

$$\sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} (C_i - C_2)^2 + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{noobj} (C_i - \hat{C}_i)^2 \quad (2.11)$$

Untuk memastikan bahwa faktor kehilangan nilai *confidence* dapat mengakibatkan pada hilangnya *bounding box* pada obyek, maka YOLO memberlakukan sebuah fungsi 1_{ij}^{obj} yang di tunjukkan pada Persaman 2.12.

$$1_{ij}^{obj} = \begin{cases} 1 & \text{apabila terdapat objek ke } i \text{ dan } j \text{ box} \\ 0 & \text{'kondisi lainnya} \end{cases} \quad (2.12)$$

Dan serupa pada kasus *downweight* pada *bounding box* dengan menerapkan fungsi 1_{ij}^{noobj} , pada Persamaan 2.13.

$$1_{ij}^{noobj} = \begin{cases} 1 & \text{apabila tidak terdapat objek ke } i \text{ pada cell} \\ 0 & \text{'kondisi lainnya} \end{cases} \quad (2.13)$$

2. Localization loss

YOLO menggunakan algoritma *sum square error* (SSE), algoritma tersebut juga memiliki kesalahan yaitu dalam pendeteksian *bounding box* yang berukuran besar atau kecil yang tidak ideal karena penyimpangan yang menonjol, untuk mengatasinya YOLO menerapkan akar kuadrat dari koordinat

tinggi dan lebar secara langsung dengan melakukan pengaturan pada parameter $\lambda_{coord} = 5$, menggunakan Persamaan 2.14.

$$\begin{aligned} & \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbf{1}_{ij}^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] \\ & + \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbf{1}_{ij}^{obj} [(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2] \end{aligned} \quad (2.14)$$

3. Classification loss

Tahap terakhir untuk mengatasi *classification loss*, YOLO menggunakan SSE untuk membandingkan kondisi probabilitas pada semua kelas agar dapat menghilangkan kesalahan kalsifikasinya pada obyek yang ada pada *grid cell* saja. YOLO melakukan pengaturan parameter $\mathbf{1}_i^{obj}$ menggunakan Persamaan 2.15.

$$\begin{aligned} \mathbf{1}_{ij}^{noobj} &= \begin{cases} 1 & \text{apabila tidak terdapat objek ke } i \text{ pada cell} \\ 0 & \text{'kondisi lainnya} \end{cases} \\ \sum_{i=0}^{S^2} \mathbf{1}_i^{obj} \sum_{c \in \text{classes}} (p_i(C) - \hat{p}_i(c))^2 \end{aligned} \quad (2.15)$$

Selengkapnya setiap fungsi dinyatakan pada Persamaan 2.16. berikut:

$$\begin{aligned} & \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbf{1}_{ij}^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] \\ & + \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbf{1}_{ij}^{obj} [(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2] \\ & + \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbf{1}_{ij}^{obj} (C_i - \hat{C}_i)^2 \\ & + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbf{1}_{ij}^{noobj} (C_i - \hat{C}_i)^2 \\ & + \sum_{i=0}^{S^2} \mathbf{1}_i^{obj} \sum_{c \in \text{classes}} (p_i(C) - \hat{p}_i(c))^2 \end{aligned} \quad (2.16)$$

2.2.24. Support Vector Regression (SVR)

Support vector regression merupakan algoritma yang menangani permasalahan regresi dan dapat menangani *overfitting* yang bertujuan untuk meminimalkan nilai *error* untuk prediksi. SVR merupakan pengembangan *Support Vector Machine* (SVM) untuk permasalahan analisis regresi yang bekerja dengan cara mencari *hyperplane* dengan margin terbesar (Risky dkk., 2014). Data hasil dari pengolahan berupa nilai bilangan real atau bilangan kontinu. Agar data sesuai dengan tipe data, SVR menerapkan pemilihan margin terbesar untuk mendapatkan nilai terbaik yang disebut dengan *epsilon-tube* (ϵ -tube) atau jarak epsilon tentunya mempertimbangkan tingkat kesalahan pada kompleksitas model (Hong., 2013) sesuai Persamaan 2.17.

$$f(x) = w^T \varphi(x) + b \quad (2.17)$$

Apabila data yang dimiliki adalah $\{(x_1, y_1), \dots, \dots, (x_\ell, y_\ell)\}$ $x_1 \in \mathcal{X} \subseteq \mathbb{R}^d$, nilai \mathcal{X} adalah sebuah ruang dari pola input ($\mathcal{X} = \mathbb{R}^d$). Dengan algoritma tersebut yang ingin dicapai adalah menemukan fungsi $f(x)$ sesuai Persamaan 2.18.

$$f(x) = w^T \varphi(x) + b \text{ dengan } w \in \mathcal{X}, b \in \mathbb{R} \quad (2.18)$$

Operasi $w^T \varphi(x)$ menunjukkan operasi *dot product* pada \mathcal{X} , pada persamaan 2.18 menjelaskan bahwa *flatness* dilakukan untuk mencari nilai w , salah satu caranya adalah dengan meminimalkan nilai ukuran norm (fungsi yang mengukur panjang vektor). Contohnya adalah $\|w\|^2 \langle w, w \rangle$ sehingga dapat disebut permasalahan optimasi vektor cembung melalui Persamaan 2.19.

$$\text{minimize } \frac{1}{2} \|w\|^2, \text{ tergantung pada } \begin{cases} y_i - w^T \varphi(x) - b \leq \epsilon \\ w^T \varphi(x) + b - y_i \leq \epsilon \end{cases} \quad (2.19)$$

Secara tidak langsung persamaan 2.18 membangun asumsi fungsi f yang mendekati nilai (x_1, y_1) dengan nilai presisi epsilon (ϵ). Dengan kata lain permasalahan optimasi vektor cembung dianggap layak dan pada kasus yang lain justru sedikit nilai *error* diperbolehkan. *Soft margin loss function* yang digunakan

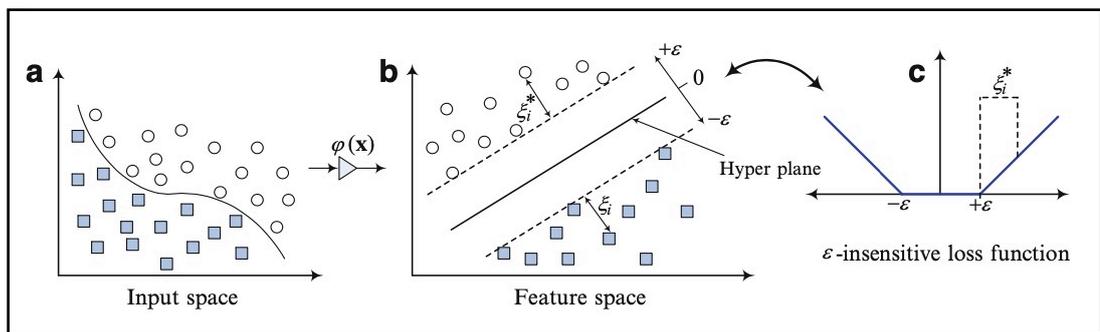
pada mesin *support vector* yang diperkenalkan yaitu memperkenalkan variabel ξ_i, ξ_i^* untuk mengatasi kendala optimasi pada Persamaan 2.20. (Cortes dan Vapnik, 1995).

$$\begin{aligned} & \text{minimize } \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{\ell} (\xi_i + \xi_i^*) & (2.20) \\ & \text{tergantung pada } \begin{cases} y_1 - \langle w, x_i \rangle - b \leq \varepsilon + \xi_i \\ \langle w, x_i \rangle + b - y_i \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases} \end{aligned}$$

Nilai konstan $C > 0$ menentukan keseimbangan yang dicapai antara fitur yang diinginkan tetapi tidak kompatibel karena *flatness* pada f memiliki nilai penyimpangan ε dapat ditoleransi. Hal ini sering disebut dengan fungsi ε -insensitive loss $|\xi|_\varepsilon$ sesuai Persamaan 2.21.

$$|\xi|_\varepsilon := \begin{cases} 0, & \text{jika } |\xi| \leq \varepsilon \\ |\xi| - \varepsilon, & \text{Lainnya} \end{cases} \quad (2.21)$$

Selanjutnya apabila digambarkan secara visual dengan model grafis akan berbentuk seperti pada Gambar 2.5.



Gambar 2.5. *Soft margin loss SVR* (Hong, 2013)

Poin x pada daerah luar margin hyperplane akan dilakukan operasi linier menggunakan Persamaan 2.18, dalam banyak kasus persamaan tersebut berfungsi dengan baik. Persamaan 2.19 formulasi ganda akan berfungsi untuk memperluas *support vector* ke fungsi non linear.

2.2.24.1 Fungsi *Kernel*

Fungsi *kernel* pada metode SVR merupakan cara untuk mengatasi permasalahan tidak linier. Dengan menggunakan *kernel* data x (*input space*) dipetakan pada sebuah *feature space* dalam dimensi yang lebih tinggi (Risky dkk., 2014). Jenis *kernel* yang digunakan pada metode SVR (Abdul Haris., 2018).

Kernel Linear ditunjukkan oleh Persamaan 2.22.

$$K(\vec{x}_i, \vec{x}_j) = \vec{x}_i, \vec{x}_j \quad (2.22)$$

Kernel Polynomial ditunjukkan oleh Persamaan 2.23.

$$K(\vec{x}_i, \vec{x}_j) = (\vec{x}_i, \vec{x}_j + 1)^p \quad (2.23)$$

Kernel Radial Basis ditunjukkan oleh Persamaan 2.24.

$$K(\vec{x}_i, \vec{x}_j) = \exp\left(-\frac{\|\vec{x}_i - \vec{x}_j\|^2}{2\sigma^2}\right) \quad (2.24)$$