BAB II
TINJAUAN PUSTAKA

2.1 Kajian Terkait

Untuk mengetahui jenis kendaraan dari sebuah citra dibutuhkan metode untuk mendeteksi dan menentukan jenis kendaraan. Pendeteksian objek dapat dilakukan dengan metode *background subtraction* dan pengaplikasian transformasi morfologi yang kemudian dengan mengambil blob *contour* objek dari citra hasil. Sedangkan untuk menentukan jenis kendaraan dari objek yang dideteksi dapat dilakukan klasifikasi objek dengan penerapan metode *Haar cascade classifier*.

Dalam mendeteksi objek pada sebuah citra maka citra harus diproses dengan *background subtraction* terlebih dahulu yaitu dimana nilai citra saat ini akan dipisah antara citra *background* dengan citra *foreground*. Citra *foreground* yang merupakan objek terdeteksi kemudian diproses agar menghasilkan blob objek kendaraan. Agar blob objek kendaraan dapat diolah dengan baik maka mesti dilakukan transformasi morfologi pada citra tersebut. Blob objek kendaraan dari tiap frame video citra jalan akan dihubungkan sehingga menghasilkan garis pergerakan kendaraan.

Pada tugas klasifikasi objek digunakan metode *Haar cascade classifier* yang merupakan modifikasi dari sistem *face detection* dalam (Viola dan Jones 2004). Metode ini diterapkan untuk mengenali objek berdasarkan nilai sederhana dari fitur citra, tetapi bukan merupakan menggunakan nilai piksel dari citra objek tersebut sehingga memberikan kelebihan yaitu komputasinya sangat cepat karena bergantung hanya pada jumlah piksel dalam persegi fitur, tidak setiap nilai piksel dari sebuah citra (Viola dan Jones 2001). Metode ini merupakan metode yang efektif dalam pendeteksian objek, dan menjadi sebuah metode utama untuk mendeteksi objek dalam (Ulfa dan Widyantoro 2018) hingga cocok untuk digunakan pada penelitian ini yaitu untuk mengetahui jenis kendaraan berupa mobil dan sepeda motor.

Penelitian serupa telah dilakukan sebelumnya, pada (Sinaga et al. 2017) digunakan metode *background subtraction* dan *haar cascade classifier* untuk

mendeteksi wajah manusia dalam tugas *people counter*. Pada (Ramadhani, Minarno, dan Cahyono 2017) , dilakukan pendeteksian mobil dengan menggunakan metode Haar cascade classifier yang dilatih menggunakan 1000 gambar mobil dan menghasilkan tingkat deteksi tertinggi sebesar 0.975 dan tingkat ketepatan klasifikasi mobil rata-rata sebesar 0.978. Pada (Lazaro, Buliali, dan Amaliah 2017) juga dilakukan penelitian menggunakan *Haar cascade classifier* untuk mendeteksi mobil pada 3 kondisi berbeda yaitu jalanan sepi, sedang dan padat dengan akurasi masing-masing 77.8%, 47.5% dan 28.2%.

2.2 Kendaraan

Menurut (KBBI Daring 2016) kendaraan merupakan sesuatu yang digunakan untuk dikendarai atau dinaiki. Sedangkan kendaraan bermotor adalah kendaraan yang menggunakan mesin (motor) untuk menjalankannya. Contoh kendaraan bermotor adalah mobil, sepeda motor, truk dan bis.

Dalam penelitian ini, penulis melakukan deteksi terhadap kendaraan bermotor yaitu mobil dan sepeda motor. Karena jalan utama di Universitas Tanjungpura adalah jalan dua jalur maka pada penelitian ini akan mendeteksi dan menghitung kendaraan keluar dan masuk ke Universitas Tanjungpura. Berikut adalah contoh citra mobil dan sepeda motor.

Gambar II.1 Sepeda motor tampak depan (kendaraan masuk).

Gambar II.2 Sepeda motor tampak belakang (kendaraan keluar).

Gambar II.3 Mobil tampak depan (kendaraan masuk).

Gambar II.4 Mobil tampak belakang (kendaraan keluar).

2.3 Trafik Kendaraan

Trafik kendaraan atau arus lalu-lintas kendaraan adalah gerak kendaraan sepanjang jalan. Arus lalu-lintas pada suatu jalan raya diukur berdasarkan jumlah kendaraan yang melewati titik tertentu selama waktu tertentu. Dalam beberapa hal lalu-lintas dinyatakan dengan Lalu-lintas Harian Rata-rata (LHR) bila periode pengamatannya kurang dari satu tahun.

Dalam (Direktorat Jenderal Bina Marga 1997) ,definisi arus lalu lintas adalah jumlah kendaraan bermotor yang melewati suatu titik jalan per satuan waktu, dinyatakan dalam kend/jam ($Qkend$), smp/jam ($Qsmp$) atau lalu-lintas harian rata-rata tahunan ($Q\ LHRT$).

2.4 Pengenalan Objek

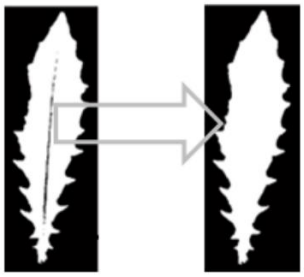
Object recognition atau pengenalan objek merupakan suatu tantangan dan juga tujuan utama dalam pengolahan citra. Tantangan yang dihadapi dalam pengenalan objek adalah bagaimana komputer dapat mengidentifikasi objek meskipun terdapat variasi dari tampilan objek tersebut. Variasi-variasi tersebut dapat berupa sudut pandang objek yang berubah-ubah, variasi cahaya atau iluminasi, dan perbedaan pose objek terhadap penangkap gambar.

Pengenalan objek telah dikembangkan dengan beberapa metode berdasarkan pendekatan apa yang digunakan untuk mengidentifikasi suatu objek yaitu berdasarkan bentuk geometri *(shape/geometry-based approaches*), tampilan (*appearance-based*), dan berdasarkan fitur. Metode yang digunakan pada penelitian ini adalah *Haar Cascade Classifier*. *Haar Cascade Classifier* merupakan metode berdasarkan fitur yaitu fitur Haar-*like*.

2.5 Transformasi Morfologi

Menurut (Kadir dan Susanto 2013), transformasi morfologi merupakan operasi yang umum digunakan pada citra biner (hitam-putih) untuk mengubah struktur bentuk objek yang terkandung dalam citra. Sebagai contoh, lubang pada daun dapat ditutup melalui operasi morfologi sebagaimana ditunjukkan pada gambar II.5. Objek-objek daun yang saling berimpitan pun dapat dipisahkan melalui morfologi, sebagaimana ditunjukkan pada Gambar II.6. Beberapa contoh lain aplikasi morfologi adalah sebagai berikut.

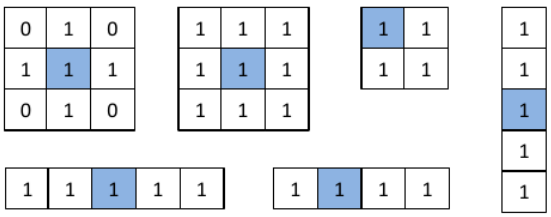
1. Membentuk filter spasial.
2. Memperoleh skeleton (rangka) objek.
3. Menentukan letak objek di dalam citra.
4. Memperoleh bentuk struktur objek.



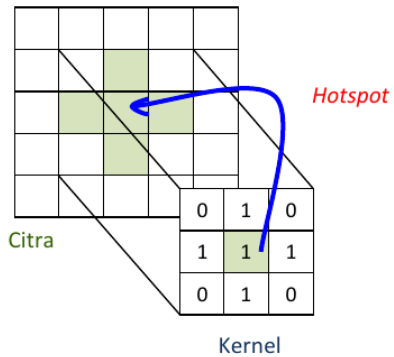
Gambar II.5 Tulang daun dapat dianggap sebagai bagian daun melalui morfologi (Kadir dan Susanto 2013)

Gambar II.6 Daun-daun yang bersinggungan dapat dipisahkan melalui morfologi, yang memperkecil ukurannya (Kadir dan Susanto 2013)

Pada *sequence to sequence model* ini inti operasi morfologi melibatkan dua larik piksel. Larik pertama berupa citra yang akan diterapkan operasi morfologi, sedangkan larik kedua dinamakan sebagai kernel atau *structuring element* (elemen penstruktur) (Shih, 2009 dalam (Kadir dan Susanto 2013)). Contoh kernel ditunjukkan pada Gambar II.7 Pada contoh tersebut, piksel pusat (biasa diberi nama *hotspot*) ditandai dengan warna abu-abu. Piksel pusat ini yang menjadi pusat dalam melakukan operasi terhadap citra, sebagaimana diilustrasikan pada Gambar II.10.



Gambar II.7 Contoh beberapa kernel (Kadir dan Susanto 2013)



Gambar II.8 Operasi kernel terhadap citra (Kadir dan Susanto 2013)

Dua operasi yang mendasari morfologi yaitu dilasi dan erosi. Dua operasi

lain yang sangat berguna dalam pemrosesan citra adalah *opening* dan *closing* dibentuk melalui dua operasi dasar itu.

Erosi dapat diperoleh dengan melebarkan komplemen dari piksel hitam dan kemudian mengambil komplemen dari *set point* yang dihasilkan. Erosi merupakan proses penghapusan titik-titik batas objek menjadi bagian dari latar, berdasarkan structuring element yang digunakan. Pada operasi ini, ukuran objek diperkecil dengan mengikis sekeliling objek. Ada dua cara yang dapat dilakukan untuk melakukan proses erosi, yaitu sebagai berikut.

1. Dengan mengubah semua titik batas menjadi titik latar.
2. Dengan mengatur semua titik di sekeliling titik latar menjadi titik latar.

$$E(A,B) = A \ominus (-B) = \bigcup_{\beta \in B}(A - \beta) \quad (2.1)$$

Operasi morfologi dasar matematika erosi dilakukan berdasarkan aljabar Minkowski pada persamaan 2.1.

$$-B = \{-B | \beta \in B\} \quad (2.2)$$

Dengan persamaan 2.2, himpunan A atau B dapat dianggap sebagai "citra". A biasanya dianggap sebagai citra dan B disebut *structuring element*. Dilasi adalah transformasi morfologi yang menggabungkan dua himpunan dengan menggunakan penjumlahan vektor elemen himpunan. Dilasi merupakan proses penggabungan titik-titik latar menjadi bagian dari objek, berdasarkan *structuring element* yang digunakan. Proses ini adalah kebalikan dari erosi, yaitu mengubah latar di sekeliling objek menjadi bagian dari objek tersebut (Anne, 2011). Ada dua cara untuk melakukan operasi ini, yaitu sebagai berikut.

1. Dengan mengubah semua titik latar yang bertetangga dengan titik batas menjadi titik objek.
2. Dengan mengubah semua titik di sekeliling titik batas menjadi titik objek.

Operasi morfologi dasar matematika dilasi dilakukan berdasarkan aljabar Minkowski pada persamaan 2.3.

$$D(A,B) = A \oplus B = \bigcup_{\beta \in B}(A + \beta) \quad (2.3)$$

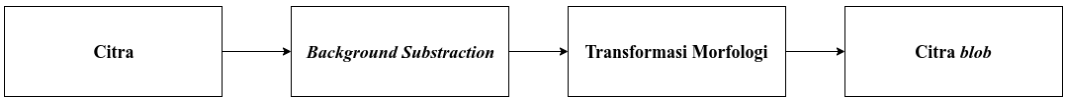
Operasi opening adalah operasi erosi yang diikuti dengan dilasi dengan menggunakan elemen penstruktur yang sama. Operasi ini berguna untuk menghaluskan kontur objek dan menghilang kan seluruh piksel di area yang

terlalu kecil untuk ditempati oleh elemen penstruktur. Dengan kata lain, semua struktur latar depan yang berukuran lebih kecil daripada elemen penstruktur akan tereliminasi oleh erosi dan kemudian penghalusan dilakukan melalui dilasi (Kadir dan Susanto 2013)

Operasi closing berguna untuk menghaluskan kontur dan menghilangkan lubang-lubang kecil. Operasi closing dilaksanakan dengan melakukan operasi dilasi terlebih dahulu dan kemudian diikuti dengan operasi erosi (Kadir dan Susanto 2013)

2.6 Blob Detection

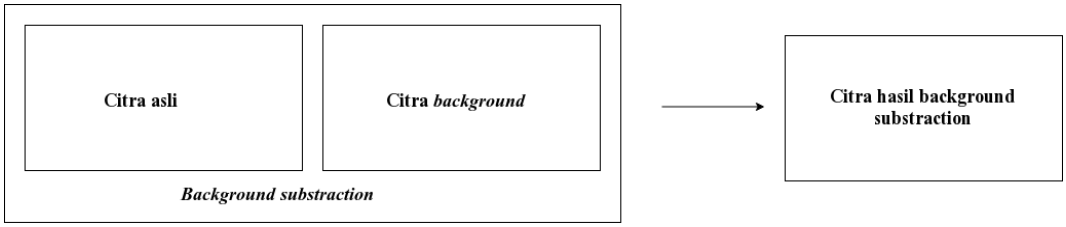
Algoritma pengolahan blob digunakan untuk menentukan suatu grup dari piksel saling berhubungan satu sama lain atau tidak. Metode ini sangat berguna untuk mengidentifikasi objek yang terpisah-pisah pada suatu citra atau menghitung jumlah dari suatu objek pada suatu citra. Pada pendeteksian blob, citra terlebih dahulu diproses menggunakan *background subtraction* dan *thresholding* hingga menjadi citra dengan tampilan hitam putih (biner). Pada citra hasil akan terlihat blob objek kendaraan yang dideteksi. Setelah itu, citra ini diproses dengan transformasi morfologi untuk memperjelas blob yang telah dideteksi. Pada Gambar II.9 digambarkan skema proses untuk mendapatkan citra blob.



Gambar II.9 Skema deteksi blob.

2.7 Background Subtraction

Background adalah bagian gambar yang tidak banyak berubah dari serangkaian gambar bergerak. *Background* dapat diperoleh dengan cara mengambil nilai rata-rata dari serangkaian gambar, nilai rata-rata tersebut akan mendekati gambar latar belakang yang diinginkan (Hartoto 2011)



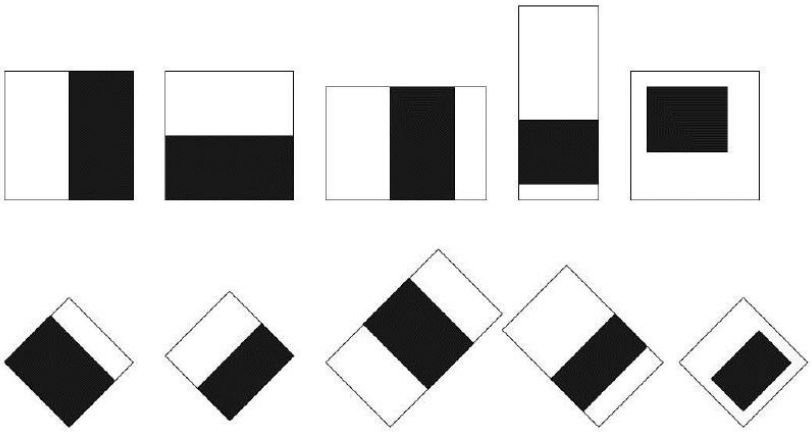
Gambar II.10 Skema *background subtraction*.

Prinsip dasar deteksi gerakan (*motion detection*) adalah membandingkan antara dua buah citra $f(x, y, t1)$ dan $f(x, y, t2)$ sehingga menghasilkan citra baru $r(x, y)$ yang memiliki nilai 0 (hitam) atau 1 (putih) dengan kriteria seperti pada persamaan 2.4 (Hartoto, 2011).

$$r(x,y) = \begin{Bmatrix} 1, \text{jika} \mid f(x,y,t_1) - f(x,y,t_2) \mid < T \\ 0, \text{untuk nilai lainnya} \end{Bmatrix} \quad (2.4)$$

2.8 Haar Cascade Classifier

Haar-like feature pertama kali diusulkan oleh Paul Viola dan Michael Jones (Viola dan Jones 2001) untuk keperluan mendeteksi wajah manusia (Viola dan Jones 2004). Metode ini kemudian diperbarui kembali oleh Rainer Lienhart dan Jochen Maydt (Lienhart dan Maydt 2002). Ide dari *Haar-like feature* adalah sebuah *classifier* yang di-*training* dengan sejumlah sampel citra dari suatu objek. Classifier di-*training* menggunakan algoritma Adaboost. Dalam kasus ini, sampel citra yang digunakan adalah citra dari kendaraan, berupa kendaraan tampak depan dan belakang. Ukuran citra yang digunakan untuk training harus sama (misalkan $20x20$), di mana nantinya ini akan menjadi sampel positif. Sampel negatif adalah citra dari objek yang berbeda-beda namun tetap memiliki ukuran yang sama. Kumpulan citra ini akan menghasilkan kumpulan fitur objek yang disebut sebagai *cascade*. *Classifier* akan menghasilkan nilai "1" jika pada citra yang dimasukkan terdapat objek yang dikenali dan nilai "0" jika tidak ada objek yang dikenali.



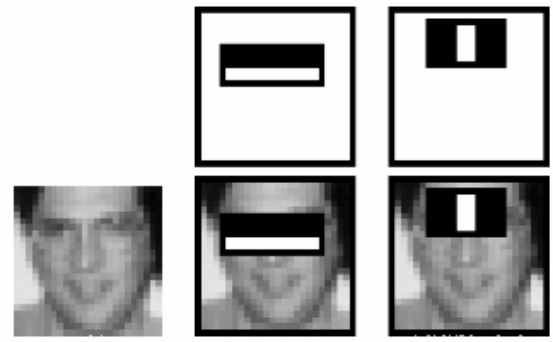
Gambar II.11 Contoh bentuk *haar-like feature*.

Hasil penerapan masing-masing fitur pada suatu wilayah gambar tertentu dihasilkan melalui jumlah piksel yang terletak di dalam segi empat hitam dari fitur yang dikurangkan oleh jumlah piksel yang overlap dengan segi empat putih. Segi empat ini didefinisikan lewat koordinat kiri atas x, y, lebar w, dan tinggi h. Total piksel yang berada dalam area segi empat ri direpresentasikan oleh $RecSum(ri)$ (Zulfikri, Yudaningtyas, dan Rahmadwati 2019)

$$feature_1 = \sum_{i=1}^{N} W_t \times RecSum(r_t) \quad (2.5)$$
$$= \sum_{i=1}^{N} W_t \times RecSum(x, y, w, h)$$

Di mana dalam persamaan 2.5 ini, nilai N, Wi, dan ri dipilih secara acak, tergantung objek yang diidentifikasi.

Dalam mengenali suatu objek, cascade akan mengabaikan area citra yang tidak memiliki objek yang memenuhi kriteria. Ini sangat membantu dalam meningkatkan performa classifier tersebut. Efeknya, kemungkinan kesalahan deteksi objek berkurang dan akurasi deteksi menjadi meningkat.



Gambar II.12 *Haar-like feature* pada tugas *face recognition.*

2.9 Python

Python adalah bahasa pemrograman interpretatif yang dianggap mudah dipelajari serta berfokus pada keterbacaan kode. Dengan kata lain, Python diklaim sebagai bahasa pemrograman yang memiliki kode-kode pemrograman yang sangat jelas, lengkap, dan mudah untuk dipahami.

Python secara umum berbentuk pemrograman berorientasi objek, pemrograman imperatif, dan pemrograman fungsional. Istilah lainnya, bahasa pemrograman multi-paradigma. Python dapat digunakan untuk berbagai keperluan pengembangan perangkat lunak dan dapat berjalan di berbagai platform sistem operasi.

2.10 OpenCV

OpenCV adalah sebuah library yang berisi fungsi-fungsi pemrograman untuk teknologi computer vision secara real time. OpenCV bersifat *open source*, bebas digunakan untuk hal-hal yang bersifat akademis dan komersial. Di dalamnya terdapat interface untuk C++, C, Python, dan Java yang dapat berjalan pada Windows, Linux, Android, dan Mac. Terdapat lebih dari 2500 algoritma dalam OpenCV. Penggunaannya antara lain pada seni interaktif, inspeksi tambang, menampilkan peta di web melalui teknologi robotik.

Fitur-fitur yang terdapat pada OpenCV antara lain: Manipulasi data gambar (alokasi, rilis, duplikasi, pengaturan, konversi), gambar dan I/O video (masukan berbasis file dan kamera, keluaran berkas gambar/video), manipulasi matriks dan vektor serta aljabar linear (produk, solusi, eigenvalues, SVD), beragam struktur data dinamis (daftar, baris, grafik), dasar pengolahan citra (filter,

deteksi tepi, deteksi sudut, pengambilan sampel dan interpolasi, konversi warna, operasi morfologi, histogram), analisis struktur (komponen yang berhubungan, pengolahan kontur, transformasi jarak, variasi momen, transformasi Hough, perkiraan poligonal, menyesuaikan garis, delaunay *triangulation*, kalibrasi kamera (menemukan dan menelusuri pola kalibrasi, kalibrasi, dasar estimasi matriks, estimasi homografi, korespondensi stereo), analisis gerakan (*optical flow*, segmentasi gerakan, penelusuran), pengenalan objek (metode eigen, HMM), dasar *Graphical User Interface* atau GUI (menampilkan gambar/video, penanganan *mouse* dan *keyboard*, *scroll-bars*), pelabelan gambar (garis, poligon, gambar teks).

Modul-modul yang terdapat pada OpenCV antara lain: cv – fungsi utama OpenCV, cvaux – fungsi pembantu OpenCV, cxcore – pendukung struktur data dan aljabar linear, highgui – fungsi GUI.

2.11 Raspberry Pi

Raspberry Pi adalah modul mikro komputer yg mempunyai *input output digital port* seperti pada *board* mikrokontroler. Tetapi jika dibandingkan *board* Raspberry Pi dan mikrokontroler yang lain, Raspberry Pi memiliki *port* untuk display berupa TV atau monitor PC serta koneksi USB untuk *keyboard* serta *mouse* yang tidak dimiliki oleh mikrokontroler jenis lain. Raspberry Pi merupakan komputer dalam satu *single board*. *Operating System* (OS) pada Raspberry Pi adalah Linux. Raspberry Pi memiliki beberapa seri seperti Raspberry Pi 1, 2, 3, 4, model A, model A+, model B dan model B+. Seri yang akan digunakan dalam penelitian ini adalah Raspberry Pi 3 model B+. Berikut ini adalah spesifikasi dari Raspberry Pi 3 model B+:

1. *System on a Chip* (SoC): Broadcom BCM2837B0 quad-core A53 (ARMv8) 64-bit @ 1.4GHz.
2. GPU: Broadcom Videocore-IV.
3. RAM: 1GB LPDDR2 SDRAM.
4. Jaringan: Gigabit Ethernet (melalui kanal USB), 2.4GHz dan 5GHz 802.11b/g/n/ac Wi-Fi.
5. Bluetooth: Bluetooth 4.2, *Bluetooth Low Energy* (BLE).
6. Penyimpanan: Micro-SD.

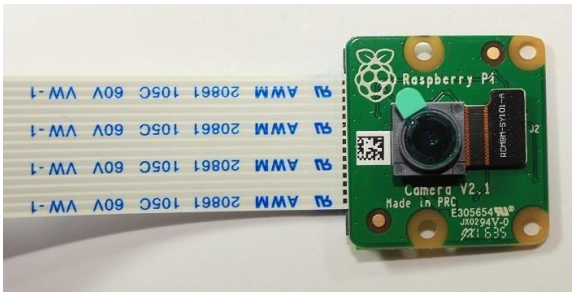
7. GPIO: 40-pin GPIO header.
8. Port: HDMI, 3.5mm audio-video jack analog, 4x USB 2.0, Ethernet, *Camera Serial Interface* (CSI), *Display Serial Interface* (DSI).
9. Ukuran dan berat: 82mm x 56mm x 19.5mm, 50g.

Gambar II.13 Raspberry Pi 3 B+.

2.12 PiCamera

Modul kamera Raspberry Pi (PiCamera) adalah produk ofisial dari Raspberry Pi Foundation. Terdapat dua jenis kamera yang telah dirilis yaitu versi pertama yang dikeluarkan pada tahun 2013 dengan resolusi 5 megapiksel dan versi kedua yang dirilis pada tahun 2016 dan memiliki resolusi 8 megapiksel.

Pada penelitian ini akan digunakan PiCamera versi kedua yang memiliki sensor gambar Sony IMX219 dengan resolusi 8-megapixel dan lensa fokus tetap. Kamera ini mampu mengambil gambar statis dengan resolusi 3280×2464 piksel dan dapat mengambil video dengan resolusi dan jumlah fps (frame per second) yaitu 1080p30, 720p60, dan 640×480p90.



Gambar II.14 PiCamera *module* v2.

2.13 SQLite

SQLite merupakan sebuah library yang mengimplementasikan mesin database SQL transaksional mandiri yang dapat digunakan serverless dan tanpa konfigurasi. Kode untuk SQLite berada dalam domain publik dan dengan demikian bebas untuk digunakan untuk tujuan apapun, komersial maupun pribadi. SQLite banyak ditemukan di berbagai aplikasi yang kita biasa gunakan, termasuk beberapa proyek high-profile.

SQLite adalah mesin database SQL yang bersifat embedded. Tidak seperti kebanyakan database SQL lainnya, SQLite tidak memiliki proses server yang terpisah. SQLite membaca dan menulis langsung ke file disk biasa. Sebuah database SQL lengkap dengan beberapa tabel, indeks, triggers, dan views, yang terkandung dalam sebuah file disk tunggal. Format file database adalah cross-platform - Anda dapat dengan bebas menyalin database antara 32-bit dan 64-bit sistem atau antara arsitektur big-endian dan little-endian. Fitur-fitur ini membuat SQLite menjadi pilihan populer pada berbagai format aplikasi. SQLite bukan sebagai pengganti Oracle tetapi sebagai pengganti fopen().

SQLite adalah library kompak. Dengan semua fitur diaktifkan, ukuran library bisa kurang dari 500 KiB, tergantung pada pengaturan platform target dan optimasi compiler. Jika fitur opsional dihilangkan, ukuran library SQLite dapat dikurangi dibawah 300 KiB. SQLite juga dapat dibuat untuk berjalan dalam ruang minimal (4 KiB) membuat SQLite pilihan mesin database populer pada gadget dengan memori terbatas seperti ponsel, PDA, dan MP3 player, namun tentu ada tradeoff antara penggunaan memori dan kecepatannya. SQLite umumnya berjalan lebih cepat semakin banyak memori yang berikan. Namun demikian, kinerja biasanya cukup baik bahkan dalam perangkat bermemori rendah.

Sebelum setiap rilis SQLite sangat hati-hati diuji sehingga memiliki reputasi sebagai library handal. Sebagian besar kode sumber SQLite dikhususkan murni untuk pengujian dan verifikasi. Sebuah test suite otomatis menjalankan jutaan uji kasus yang melibatkan ratusan juta pernyataan SQL individu dan mencapai cakupan tes 100%. SQLite merespon baik kegagalan alokasi memori dan disk I / O error. Transaksi tetap bersifat ACID bahkan jika terganggu oleh sistem crash atau gangguan listrik. Semua ini diverifikasi oleh tes otomatis

menggunakan memanfaatkan tes khusus yang mensimulasikan kegagalan sistem. Tentu saja, bahkan dengan semua pengujian ini, masih ada bug. Tapi tidak seperti beberapa proyek serupa (terutama pesaing komersial) SQLite terbuka tentang semua bug dan memberikan daftar bug termasuk daftar bug kritis dan menit-demi menit kronologi laporan bug serta perubahan kode.

2.14 Flask

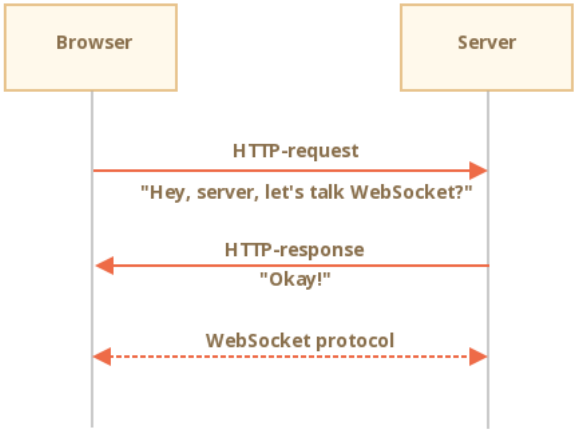
Flask adalah sebuah framework web mikro yang ditulis dalam bahasa Python. Framework ini di klasifikasi sebagai framework mikro karena framework ini tidak memerlukan tool atau library tertentu. Flask tidak memiliki pengolah database, validasi formulir, atau komponen lain dimana library biasanya sudah menyediakan. Namun, Flask mendukung ekstensi yang dapat menambahkan fitur aplikasi seolah-olah diimplementasikan di Flask itu sendiri. Ada ekstensi untuk pemetaan relasional objek, validasi formulir, penanganan unggahan, berbagai teknologi autentikasi, dan beberapa alat terkait lainnya.

2.15 Websocket

Protokol WebSocket, yang dijelaskan dalam spesifikasi RFC 6455 menyediakan cara untuk bertukar data antara browser dan server melalui koneksi yang persisten. Data dapat diteruskan di kedua arah sebagai "paket", tanpa memutus koneksi dan permintaan HTTP tambahan. WebSocket sangat bagus untuk layanan yang membutuhkan pertukaran data berkelanjutan, misal game online, sistem perdagangan real-time dan sebagainya. Ada juga protokol wss:// yang terenkripsi, serupa dengan HTTPS namun untuk soket web.

WebSocket dengan sendirinya tidak menyertakan mekanisme tingkat tinggi berupa rekoneksi, otentikasi dan banyak lainnya. Jadi dibutuhkan pustaka klien/server untuk itu, walau memungkinkan untuk mengimplementasikan kemampuan itu secara manual. Terkadang, untuk mengintegrasikan WebSocket ke dalam proyek yang ada, diperlukan untuk menjalankan server WebSocket secara paralel dengan server HTTP utama, sehingga mereka dapat berbagi suatu resource seperti database. Pada penggunaannya permintaan ke WebSocket

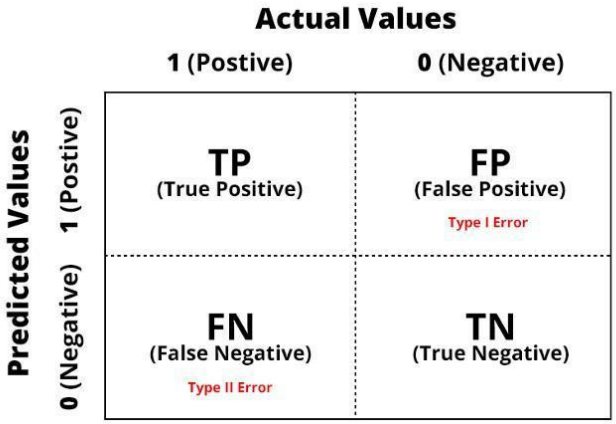
diarahkan ke wss://ws.site.com, subdomain yang mengarah ke server WebSocket, sementara https://site.com menuju server HTTP utama.



Gambar II.15 Diagram mekanisme koneksi websocket antara client dengan server.

2.16 Confusion Matrix

Confusion matrix juga sering disebut *error matrix*. Pada dasarnya *confusion matrix* memberikan informasi perbandingan hasil klasifikasi yang dilakukan oleh sistem (model) dengan hasil klasifikasi sebenarnya. *Confusion matrix* berbentuk tabel matriks yang menggambarkan kinerja model klasifikasi pada serangkaian data uji yang nilai sebenarnya diketahui. Gambar X menunjukkan confusion matrix dengan 4 kombinasi nilai prediksi dan nilai aktual yang berbeda.



Gambar II.16 *Confusion Matrix*

Terdapat 4 istilah sebagai representasi hasil proses klasifikasi pada *confusion matrix*. Keempat istilah tersebut adalah *True Positive* (TP), *True Negative* (TN), *False Positive* (FP) dan *False Negative* (FN). Berikut contoh penggunaan *confusion matrix* pada kasus sederhana untuk memprediksi seorang pasien menderita kanker atau tidak.

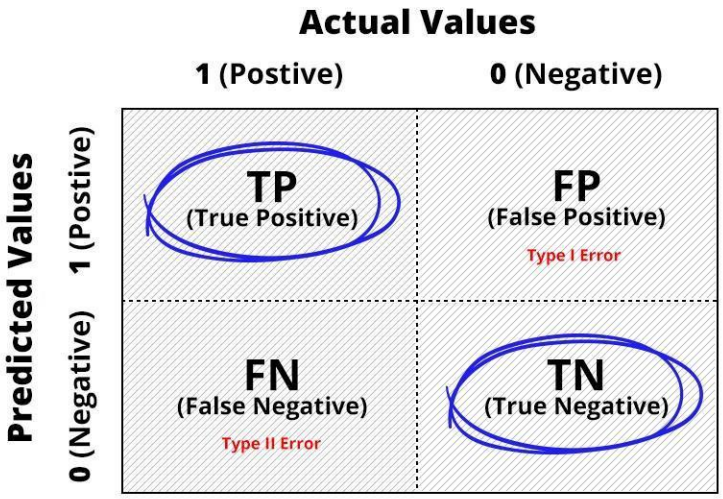
1. *True Positive* (TP)
   Merupakan data positif yang diprediksi benar. Contohnya, pasien menderita kanker (*class* 1) dan dari model yang dibuat memprediksi pasien tersebut menderita kanker (*class* 1).
2. *True Negative* (TN)
   Merupakan data negatif yang diprediksi benar. Contohnya, pasien tidak menderita kanker (*class* 2) dan dari model yang dibuat memprediksi pasien tersebut tidak menderita kanker (*class* 2).
3. *False Positive* (FP) — Type I Error
   Merupakan data negatif namun diprediksi sebagai data positif. Contohnya, pasien tidak menderita kanker (*class* 2) tetapi dari model yang telah memprediksi pasien tersebut menderita kanker (*class* 1).
4. *False Negative* (FN) — Type II Error
   Merupakan data positif namun diprediksi sebagai data negatif. Contohnya, pasien menderita kanker (*class* 1) tetapi dari model yang dibuat memprediksi pasien tersebut tidak menderita kanker (*class* 2).

Kita dapat menggunakan c*onfusion matrix* untuk menghitung berbagai *performance metrics* untuk mengukur kinerja model yang telah dibuat. Pada bagian berikut dijelaskan beberapa *performance metrics* populer yang umum dan sering digunakan: *accuracy*, *precision*, dan *recall* serta *f1-score*.

2.16.1 Akurasi

Akurasi menggambarkan seberapa akurat model dapat mengklasifikasikan dengan benar. Maka, akurasi merupakan rasio prediksi benar (positif dan negatif) dengan keseluruhan data. Dengan kata lain, akurasi

merupakan tingkat kedekatan nilai prediksi dengan nilai aktual (sebenarnya). Nilai akurasi dapat diperoleh dengan persamaan 2.6.



Gambar II.17 *Confusion matrix* yang menggambarkan nilai akurasi.

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{2.6}$$

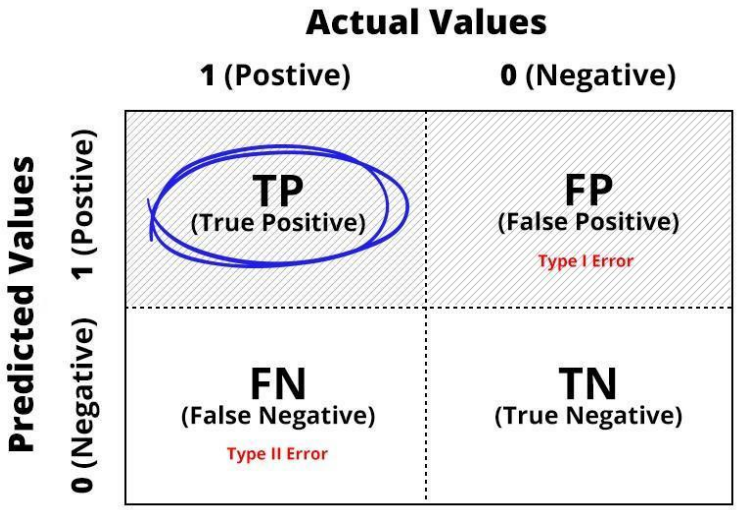
Dari contoh *confusion matrix* klasifikasi biner diatas maka dengan menghitung nilai akurasi dapat menjawab pertanyaan "Berapa persen pasien yang benar diprediksi menderita kanker maupun yang tidak menderita kanker dari keseluruhan pasien?"

$$\begin{aligned} accuracy &= \frac{jumlah\ pasien\ diprediksi\ benar\ (kanter + tidak\ kanker)}{jumlah\ pasien\ keseluruhan} \\ &= \frac{6+9}{6+9+2+3} = \frac{15}{20} = 0{,}75 \\ &= 0{,}75 \times 100\% \\ &= 75\% \end{aligned} \tag{2.7}$$

Persamaan 1 Menghitung nilai akurasi dari contoh *confusion matrix* klasifikasi biner.

2.16.2 Presisi

Precision menggambarkan tingkat keakuratan antara data yang diminta dengan hasil prediksi yang diberikan oleh model. Maka, presisi merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positif. Dari semua kelas positif yang telah diprediksi dengan benar, berapa banyak data yang benar-benar positif. Nilai presisi dapat diperoleh dengan persamaan dibawah.



Gambar II.18 *Confusion matrix* yang menggambarkan nilai presisi.

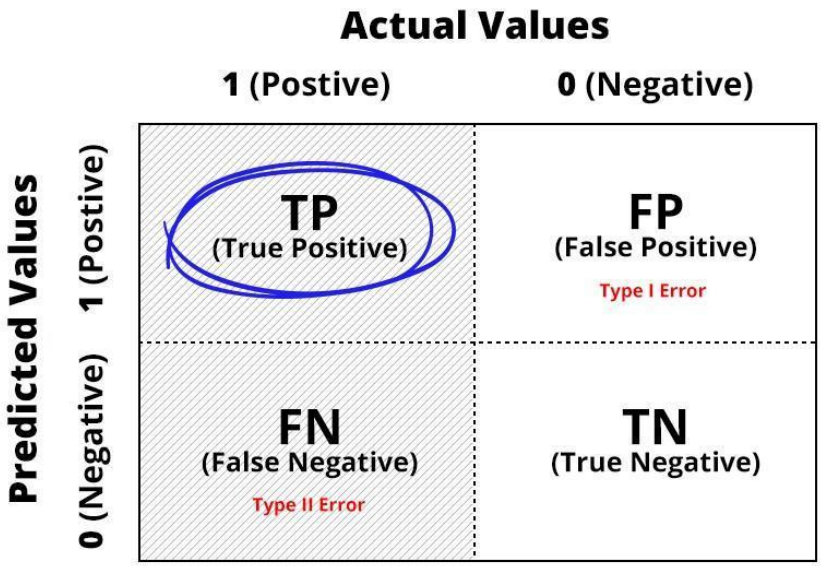
$$precision = \frac{TP}{TP+FP} \tag{2.8}$$

Dari contoh *confusion matrix* klasifikasi biner diatas maka dengan menghitung nilai presisi dapat menjawab pertanyaan "Berapa persen pasien yang benar menderita kanker dari keseluruhan pasien yang diprediksi menderita kanker?"

$$\begin{aligned} precision &= \frac{jumlah\ pasien\ diprediksi\ benar}{jumlah\ pasien\ diprediksi\ kanker} \\ &= \frac{6}{6+2} = \frac{6}{8} = 0{,}75 \\ &= 0{,}75 \times 100\% \\ &= 75\% \end{aligned} \tag{2.9}$$

2.16.3 Recall atau Sensitivity

Recall menggambarkan keberhasilan model dalam menemukan kembali sebuah informasi. Maka, *recall* merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan data yang benar positif. Nilai *recall* dapat diperoleh dengan persamaan dibawah.



Gambar II.19 Confusion matrix yang menggambarkan nilai recall.

$$recall = \frac{TP}{TP + FN} \quad (2.10)$$

Dari contoh *confusion matrix* klasifikasi biner diatas maka dengan menghitung nilai *recall* dapat menjawab pertanyaan "Berapa persen pasien yang diprediksi kanker dibandingkan keseluruhan pasien yang sebenarnya menderita kanker".

$$recall = \frac{jumlah\ pasien\ diprediksi\ benar}{jumlah\ pasien\ kanker} \quad (2.11)$$
$$= \frac{6}{6+3} = \frac{6}{9} = 0{,}66$$
$$= 0{,}66 \times 100\%$$
$$= 66\%$$