BAB 2
TINJAUAN PUSTAKA

2.1 Perangkat Keras

Sistem pendeteksian kerumunan dan masker ini memerlukan perangkat keras yang terdiri dari masukan, bagian untuk memproses dan bagian untuk mengeluarkan keluaran yang diperlukan.

2.1.1 *Mini PC*

Mini PC (*Mini Personal Computer)* adalah sebuah komputer yang dirancang dalam ukuran kecil [27]. Dipilihnya Mini PC pada sistem ini dikarenakan ukuran Mini PC yang kecil sehingga dapat memungkinkan bisa dimasukkan ke dalam sebuah robot atau dapat dikondisikan untuk berada di sekitar robot. Performa Mini PC jika dibandingkan dengan komputer pada umumnya pun tidak kalah hebatnya, lalu Mini PC menggunakan sumber daya listrik DC yang akan memudahkan jika diletakkan di dalam robot atau di sekitar robot karena dapat menggunakan catu daya baterai yang bisa diambil dari baterai yang ada pada robot. Gambar Mini PC dapat dilihat pada *Gambar 2.1*.

Gambar 2.1 Mini PC

2.1.2 Kamera

Kamera merupakan sebuah alat yang dipasang dengan lensa untuk mengambil sebuah citra digital [28]. Citra digital yang diperoleh, digunakan

pada robot untuk diproses oleh sistem, parameter yang diambil dari sebuah citra pada penelitian ini adalah bentuk objek dan jarak antara objek. Gambar Kamera dapat dilihat pada *Gambar 2.2*.

Gambar 2.2 Kamera Webcam

2.1.3 *Speaker*

Speaker merupakan sebuah alat pengeras suara [29]. Pada penelitian ini, speaker dipakai untuk mengeluarkan suara terhadap orang-orang yang melanggar protokol kesehatan seperti tidak menjaga jarak dan tidak memakai masker.

2.1.4 Gimbal *Stabilizer*

Sebuah gimbal adalah sebuah bingkai atau penyangga yang dapat berputar di sekitar satu sumbu yang memungkinkan sebuah objek terpasang di penyangga tersebut untuk berputar pada satu sumbu tersebut. Untuk mendapatkan lebih dari satu derajat kebebasan, lebih dari satu gimbal digunakan [30]. Pada penelitian ini, gimbal *stabilizer* digunakan untuk menstabilkan posisi kamera agar tidak terjadi guncangan yang mempengaruhi akurasi pendeteksian. Gambar Gimbal *Stabilizer* dapat dilihat pada *Gambar 2.3*.

Gambar 2.3 Gimbal *Stabilizer*

2.2 Robot Ikon Unikom

Robot Nakula merupakan salah satu dari robot ikon milik Universitas Komputer Indonesia, yang dikembangkan oleh Divisi Robotika Unikom. Robot tersebut dibuat dan perkenalkan pertama kali pada bulan September tahun 2017 pada acara penyambutan mahasiswa baru Unikom di gedung Sasana Budaya Ganesha (Sabuga). Robot ini merupakan robot semi humanoid yang berukuran sekitar 1.2 meter berwarna putih polet biru. Robot Nakula mempunyai kemampuan untuk bernavigasi, bersalaman, berbicara dengan manusia, dan berekspresi melalui mimik wajahnya. Dibekali dengan beberapa mikrokontroler yang difungsikan untuk bernavigasi dan berinteraksi.

Fungsi utama dari robot ini yaitu digunakan sebagai robot pentas yang biasa ditampilkan pada acara khusus Universitas Komputer Indonesia, seperti acara penyambutan mahasiswa baru dan acara wisuda. Robot ini juga pernah diundang untuk acara besar pada saat Seminar Digital & Risk Management in Insurance (DRiM) pada tahun 2019 di Nusa Dua Bali. Berikut merupakan tampilan dari salah satu robot ikon Unikom yaitu Nakula yang dapat dilihat pada *Gambar 2.4.*

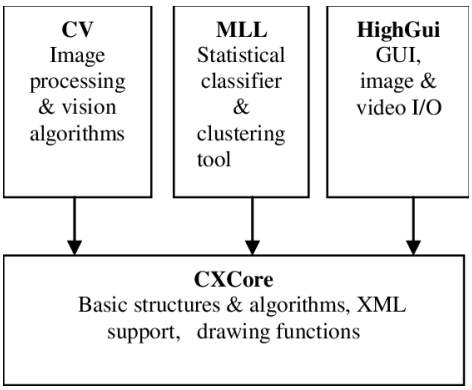
Gambar 2.4 Tampilan Robot Ikon Nakula

2.3 *Open Computer Vision*

Computer Vision merupakan salah satu dari banyak bidang dalam ilmu komputer. *Computer Vision* adalah penggunaan komputer untuk mengenali dan mengklasifikasikan isyarat visual, dengan tujuan untuk mendapatkan informasi dari gambar, *computer vision* juga bisa digunakan pada video dikarenakan video pada dasarnya hanyalah serangkaian gambar [31]. Ada beberapa cara *computer vision* dapat dicapai, dari menggunakan model matematika hingga menggunakan teknik yang lebih modern, seperti *machine learning*. Pada zaman sekarang yang serba canggih ini, *computer vision* digunakan untuk berbagai macam hal, mulai dari pengenalan dan klasifikasi [31].

Salah satu *library* yang seringkali dipakai yaitu adalah OpenCV (*Open Computer* Vision). OpenCV (*Open Computer Vision*) adalah sebuah *Library* yang sudah sangat familiar pada bidang pengolahan citra yang bersifat *open source*. *Library OpenCV* ditulis menggunakan bahasa pemrograman C dan C++ dan dapat dijalankan pada sistem operasi Linux, Windows dan MAC OS X. *Library* OpenCV memiliki lebih dari 2500 algoritma yang sangat berkaitan dengan *computer vision* dan *machine learning*. Algoritma tersebut digunakan untuk mendeteksi, mengenali dan mengklasifikasikan objek [32].

OpenCV juga bisa digunakan untuk meningkatkan kualitas dari gambar. Beberapa pengimplementasian dari OpenCV diantaranya adalah *Face Recognition*, *Face Detection*, *Object Detection* dan masih banyak yang lainnya [33]. Struktur dari OpenCV terdiri dari *basic image processing* dan algoritma *computer vision* tingkat tinggi. MLL merupakan *Machine Learning Library* yang terdiri dari *Statistical Classifiers* dan *Clustering Tools*. HighGUI terdiri dari *Input*/*Output* dan fungsi untuk menyimpan dan memanggil video dan gambar. CXCore terdiri dari struktur basis dan konten. Struktur dari OpenCV dapat dilihat pada *Gambar 2.5*.



Gambar 2.5 Struktur OpenCV [33]

2.4 Pengenalan Objek Berbasis *Computer Vision*

Computer Vision merupakan salah satu cabang dari bidang ilmu pengolahan citra (*Image Processing*) yang memungkinkan komputer dapat melihat seperti manusia. Dengan *vision* tersebut, komputer dapat mengambil keputusan, melakukan aksi dan mengenali terhadap suatu objek [33]. Pengenalan terhadap suatu objek dapat dilakukan menggunakan algoritma-algoritma yang sudah banyak digunakan, contohnya seperti *Convolutional Neural Networks (CNN)*, *Histogram of Oriented Gradients (HOG)*, *Single Shot Detector (SSD), Region-based Convolutional Neural Networks (R-CNN), Fast R-CNN, Faster R-CNN* dan yang digunakan pada penelitian ini yaitu *You Only Look Once (YOLO)*.

2.4.1 *Convolutional Neural Networks (CNN)*

Convolutional Neural Networks (CNN) adalah salah satu jenis *neural network* yang biasa digunakan pada *image processing* untuk mengklasifikasikan suatu gambar. CNN mengambil gambar dan menerapkan *weight* dan *bias* untuk mengklasifikasikan gambar dengan tepat. Pada saat ini, CNN banyak digunakan untuk mengklasifikasikan objek, mendeteksi gerakan objek pada video. CNN berisi 4 lapisan yang digunakan untuk melatih model secara efektif untuk mengklasifikasikan gambar dengan tepat [33]. Fungsionalitas dari tiap lapisan adalah sebagai berikut:

1. *Convolution Layer*

   Layer ini merupakan *layer* paling atas pada arsitektur CNN. *Layer* ini berisi berbagai macam fitur untuk memfilter gambar, seperti Sobel dan

Scharr. Ukuran dari hasil filter harus lebih kecil dari gambar masukan. Di *layer* ini dilakukan *filter* dengan menghitung nilai di antara *filter* dan gambar masukan di setiap posisi dengan menggeser di sekitar gambar masukan. Setelah mendapatkan semua nilai, nilai tersebut dijumlahkan untuk mendapatkan 1 nilai. Nilai-nilai tersebut kemudian dikirim ke *layer* berikutnya.

2. *Activation Layer*

Activation function digunakan pada *layer* ini untuk memutuskan apakah suatu *neuron* harus aktif atau dibuang di *neural network*. *Activation function* yang banyak digunakan adalah *sigmoid*, *tanh*, *relu* dan *leaky relu*.

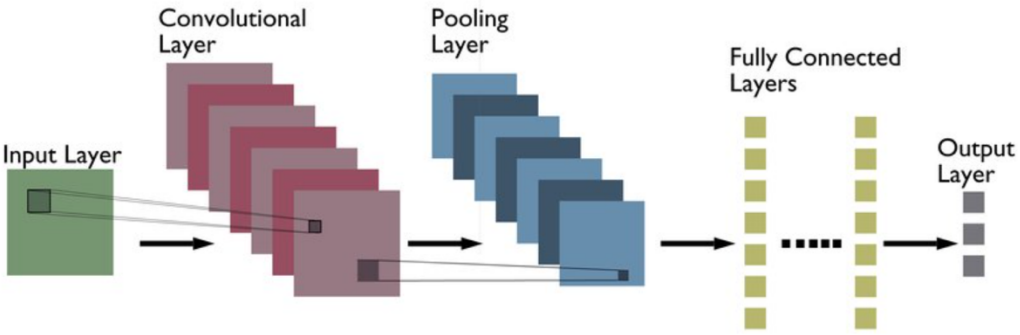
3. *Pooling Layer*

Pooling layer melakukan downsampling terhadap fitur yang diperoleh dari *layer* sebelumnya. Oleh karena itu, *layer* ini biasa dikenal sebagai *layer downsampling*. Hal tersebut mengurangi *noise* dan distorsi dengan memilih hanya fitur penting dari *layer* sebelumnya yaitu hanya memilih nilai maksimum untuk menyimpan maksimum informasi. Ada 2 jenis *pooling layer*, yaitu *Average pooling* yang digunakan untuk mengambil rata-rata dari semua nilai piksel dari setiap jendela, *Max pooling* digunakan untuk menemukan nilai maksimum dari setiap jendela sehingga dapat menyimpan informasi dari gambar secara maksimal.

4. *Fully Connected Layer*

Fully connected layer mengambil *neuron* dari lapisan sebelumnya dan terhubung ke setiap unit aktivasi *layer* berikutnya. *Layer* ini digunakan untuk mengkompilasi nilai-nilai yang diperoleh dari *layer* sebelumnya untuk memberikan hasil terakhir. Pada *layer* ini digunakanlah *activation function* seperti *sigmoid* dan *softmax* untuk mengklasifikasikan gambar menjadi kelas.

Arsitektur dari *Convolutional Neural Network (CNN)* dapat dilihat pada *Gambar 2.6*.

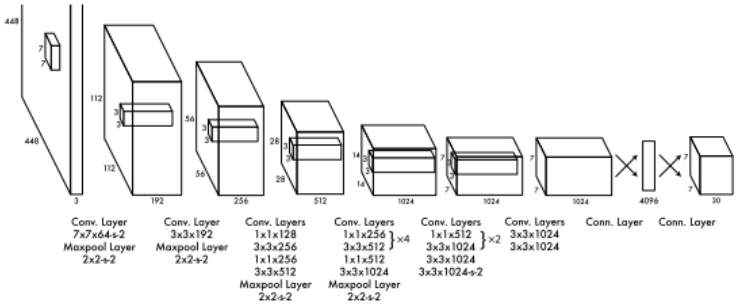


Gambar 2.6 Arsitektur CNN

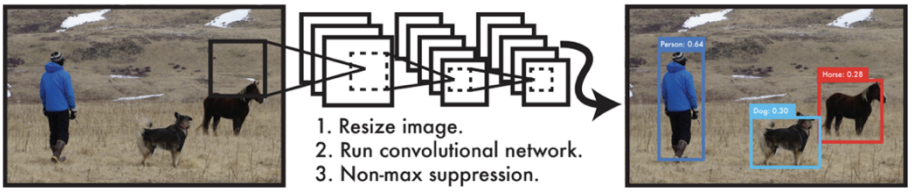
2.4.2 *You Only Look Once (YOLO)*

You Only Look Once (YOLO) merupakan sebuah algoritma yang dikembangkan untuk pendeteksian objek secara *realtime*. Pendeteksian objek tersebut dilakukan dengan menggunakan sebuah model yang diterapkan pada sebuah citra di beberapa lokasi dan skala. Daerah dengan citra yang diberi *score* tertinggi, maka akan dianggap sebuah pendeteksian [33]. Arsitektur dari *YOLO* memiliki 24 lapisan konvolusional dan diikuti 2 lapisan yang terhubung sepenuhnya. Pada lapisan di atas bergantian 1x1 lapisan konvolusional untuk mengurangi ruang fitur dari lapisan sebelumnya.

Lapisan konvolusional tersebut dilatih menggunakan model klasifikasi ImageNet, kemudian digandakan resolusinya untuk dideteksi. Sedangkan pada proses *YOLO* itu sendiri, *YOLO* memproses gambar secara sederhana dan lugas. Dimana tahap pertama yaitu dilakukan pengubahan pada ukuran gambar masukan ke ukuran resolusi 448x448, lalu tahap kedua yaitu menjalankan jaringan konvolusional tunggal yang bekerja pada gambar, lalu tahap yang terakhir yaitu membatasi deteksi yang dihasilkan sebesar keyakinan model [33]. Gambar dari arsitektur dan proses pada *YOLO* dapat dilihat pada *Gambar 2.7* dan *Gambar 2.8*.



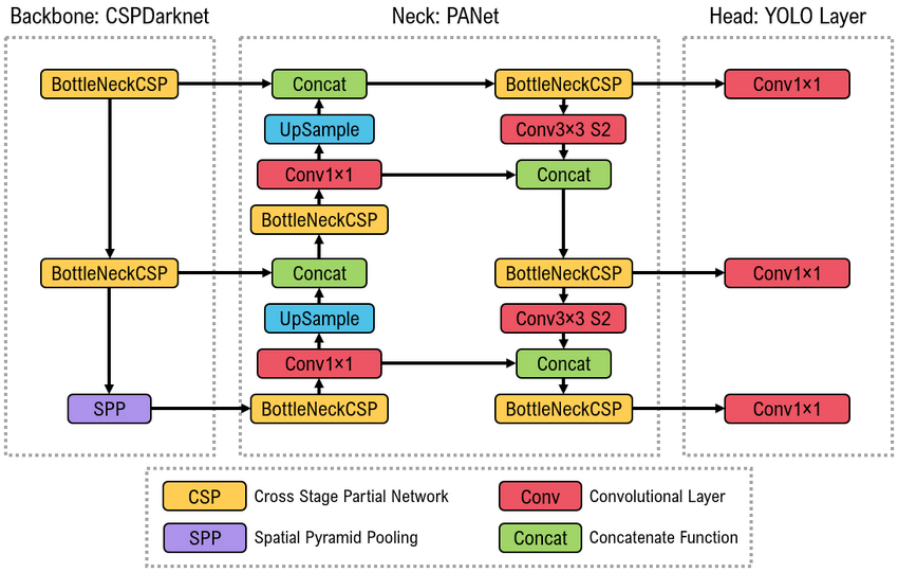
Gambar 2.7 Arsitektur Algoritma *YOLO* [33]



Gambar 2.8 Proses Algoritma *YOLO* [33]

Kelebihan dari algoritma *YOLO* jika dibandingkan dengan algoritma pendeteksi objek lainnya adalah algoritma ini memiliki kecepatan yang cepat, memiliki performa yang baik dan memiliki akurasi yang tinggi dalam mendeteksi objek secara *realtime* [34]. Sedangkan kekurangan dari algoritma *YOLO* adalah ukuran model untuk algoritma ini yang sangat besar, sehingga memerlukan spesifikasi komputer yang memiliki penyimpanan yang memadai untuk melakukan pendeteksian objek menggunakan algoritma *YOLO* ini [34].

YOLO memiliki 3 komponen utama, yaitu *backbone*, *head* dan *detection*. Bagian *backbone* adalah *Convolutional Neural Networks* (CNN) yang mengumpulkan dan membentuk fitur gambar di granularitas yang berbeda. Algoritma *YOLO* mengimplementasikan *Center and Scale Prediction* (CSP) *Bottleneck* untuk merumuskan fitur gambar. Bagian *head* adalah serangkaian lapisan untuk menggabungkan fitur gambar untuk melanjutkannya kepada proses prediksi. *YOLO* juga mengimplementasikan PA-NET untuk agregasi fitur. Bagian *detection* adalah proses yang memanfaatkan fitur dari *head* dan mengambil langkah-langkah untuk memprediksi *box* dan *class* dari suatu objek [24]. Struktur komponen dari algoritma *YOLO* dapat dilihat pada *Gambar 2.9*.



Gambar 2.9 Struktur Komponen *YOLO* [24]

2.5 *Dataset*

Dataset merupakan sebuah kumpulan data yang berasal dari informasi-informasi pada masa lalu dan siap untuk dikelola menjadi sebuah informasi baru yang biasanya digunakan untuk klasifikasi, prediksi pada *machine learning* dan deteksi objek. *Dataset* dapat berupa data apapun yang valid, bisa berupa file excel, bisa berupa gambar, bisa berupa file xml. Jenis dari *dataset* sendiri ada 2, yaitu ada *private* dan *public dataset*.

Private dataset adalah *dataset* yang dapat diambil dari sebuah organisasi yang akan dilakukan sebagai objek penelitian, seperti data bank, rumah sakit, sekolah dan lain sebagainya. Sedangkan *public dataset* adalah *dataset* yang bisa diambil dari *repository* publik yang disepakati oleh pakar peneliti pada bidang *data mining*. *Dataset* bertujuan untuk menguji suatu metode penelitian yang dikembangkan oleh para pakar peneliti dengan *public dataset* maupun *private dataset*. Contoh dari *public dataset* dan yang digunakan pada penelitian ini adalah *dataset* MS COCO dan *dataset* Kaggle.

2.5.1 *Microsoft Common Objects in Context (MS COCO)*

Microsoft Common Objects in Context atau yang disingkat dengan MS COCO adalah *dataset* yang sudah familiar didengar jika berhubungan dengan pendeteksian objek. MS COCO adalah sebuah *dataset* yang digunakan untuk

mendeteksi objek atau mengklasifikasikan objek dalam skala yang besar, segmentasi dan memberi label pada *dataset*. Fitur-fitur yang ada pada *dataset* MS COCO ini adalah segmentasi objek, memiliki 330 ribu gambar dan sudah lebih dari 200 ribu gambar yang diberi label, memiliki 80 kategori objek yang diantaranya adalah orang (*person*), mobil (*car*), pesawat (*airplane*) dan lain sebagainya.

2.5.2 Kaggle

Kaggle, anak perusahaan dari Google LLC, adalah komunitas online untuk para *data scientist* dan praktisi dari *machine learning*. Kaggle memungkinkan pengguna untuk menemukan dan menerbitkan *dataset,* menjelajahi dan membangun model dalam lingkungan *data-science* berbasis web, bekerja dengan para *data scientist* dan insinyur *machine learning*, dan mengikuti kompetisi untuk memecahkan tantangan *data science*.

Salah satu fitur unggulan yang ada pada Kaggle ini adalah terdapatnya banyak *public dataset*, contoh dari *public dataset* yang ada pada situs Kaggle ini adalah *dataset* Iris yang digunakan untuk mengklasifikasikan spesies dari tumbuhan Iris, lalu ada *dataset* diabetes yang digunakan untuk mengklasifikasikan penyakit diabetes pada seseorang, lalu *dataset* yang digunakan pada penelitian ini yaitu *dataset* pemakaian masker yang digunakan untuk dilakukannya pendeteksian pemakaian masker pada penelitian ini.

2.6 Pendeteksi Kerumunan

Kerumunan merupakan sekumpulan orang (2 orang atau lebih) yang berada pada suatu tempat yang tidak saling menjaga jaraknya satu sama lain. Menjauhi kerumunan merupakan salah satu anjuran dari *World Health Organization* (WHO) untuk mencegah penyebaran dari virus *Coronavirus Disease-19* (Covid-19) yang sedang merebak di seluruh dunia. Seiring dengan berkembangnya teknologi dan masih merebaknya virus Covid-19 di dunia ini, maka untuk menekan penyebaran dari virus ini adalah dengan diterapkannya sistem pendeteksian kerumunan.

Sistem pendeteksian kerumunan merupakan sistem untuk mendeteksi terjadinya kerumunan pada suatu tempat dengan cara mendeteksi jarak antara satu orang dengan orang yang lain. Jika terdapat dua orang atau lebih yang terdeteksi

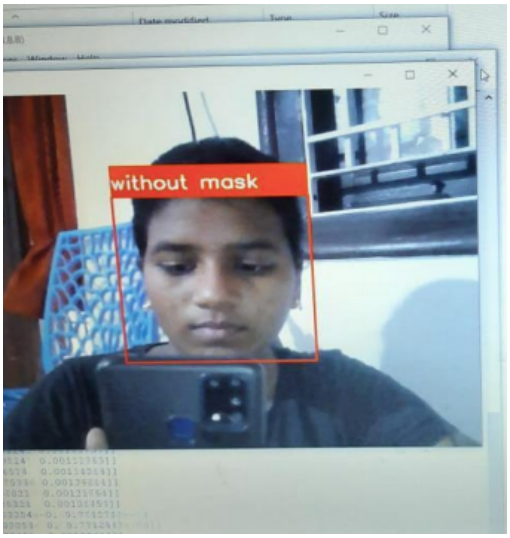
tidak menjaga jaraknya satu sama lain atau dengan kata lain jarak antara satu orang dengan orang yang lain kurang dari jarak aman yang ditentukan oleh WHO yaitu 2 meter, maka mereka bisa disebut sebagai orang yang melakukan kerumunan [20]. Contoh sistem pendeteksian kerumunan dapat dilihat pada *Gambar 2.10*.

Gambar 2.10 Contoh Pendeteksi Kerumunan [20]

2.7 Pendeteksi Penggunaan Masker

Masker merupakan alat perlindungan pernafasan yang digunakan sebagai metode untuk melindungi individu dari menghirup zat-zat bahaya atau kontaminan yang berada di udara, terutama pada saat virus Covid-19 merebak. Penggunaan masker wajah menjadi salah satu anjuran dari WHO untuk semua umat manusia dikarenakan virus Covid-19 yang menyebar melalui *droplet*. Penggunaan masker yang benar yaitu harus menutupi bagian hidung dan mulut sehingga *droplet* yang berada di udara tersaring dan terhalangi oleh masker wajah yang dipakai.

Akan tetapi, masih ada beberapa orang yang belum benar dalam hal penggunaan masker, terutama saat berada di lingkungan publik. Maka dari itu, hadirlah sistem pendeteksian penggunaan masker untuk mendeteksi orang-orang yang melanggar protokol kesehatan penggunaan masker, dari mulai mereka tidak memakai masker, hingga pemakaian maskernya tidak benar (tidak menutupi mulut atau tidak menutupi hidung) [15]. Contoh sistem pendeteksi penggunaan masker dapat dilihat pada *Gambar 2.11*.



Gambar 2.11 Contoh Pendeteksi Penggunaan Masker [15]

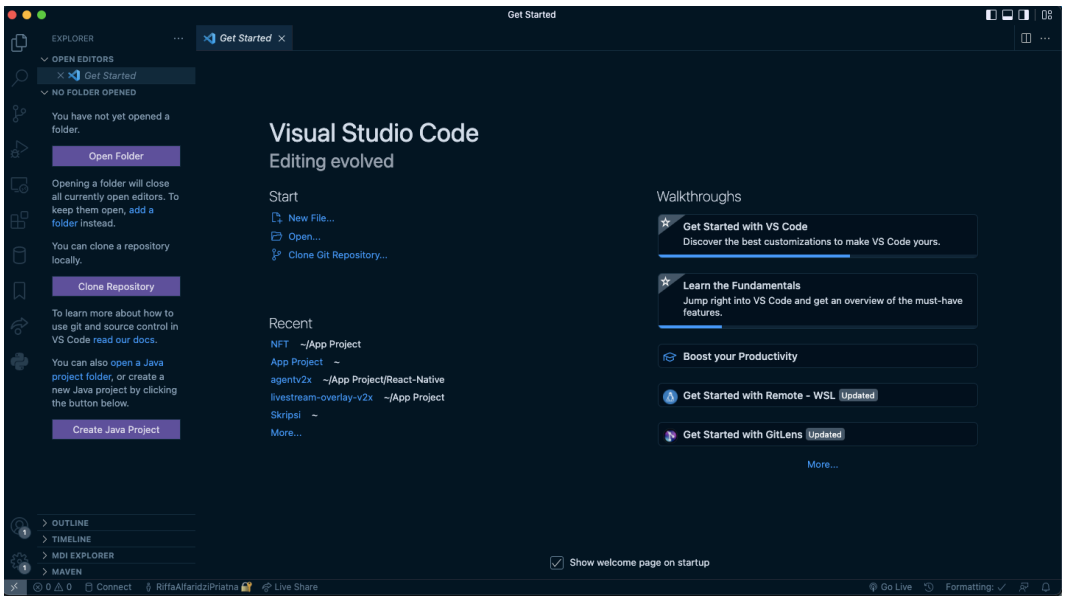
2.8 Perangkat Lunak

Dalam proses pembangunan sistem pendeteksian kerumunan dan masker ini, penulis menggunakan beberapa software bantuan, yang diantaranya adalah sebagai berikut.

2.8.1 Visual Studio Code

Visual Studio Code merupakan sebuah aplikasi *code editor* yang bisa dijalankan di perangkat desktop berbasis Windows, Linux dan MacOS. Aplikasi ini dikembangkan oleh salah satu raksasa teknologi dunia, yaitu Microsoft. Visual Studio Code merupakan aplikasi editor yang cukup powerful tapi tetap ringan ketika digunakan untuk membuat dan mengedit *source code* berbagai bahasa pemrograman. Misalnya seperti Node.js, JavaScript, TypeScript dan masih banyak yang lainnya. Bahkan sekarang Visual Studio Code juga sudah kompatibel dengan bahasa dan *runtime environment* yang lain, seperti PHP, Python, Java dan .NET.

Hal tersebut dapat terwujud berkat ekosistemnya yang luas dan ketersediaan *extension* yang melimpah. Pada aplikasi Visual Studio Code juga terdapat berbagai fitur yang mendukung para pengembang aplikasi/sistem untuk membantu pengembangan, seperti fitur *code intellisense* yang berguna untuk menyelesaikan *syntax* secara otomatis dan juga *code refactoring*. Tampilan dari Visual Studio Code dapat dilihat pada *Gambar 2.12*.

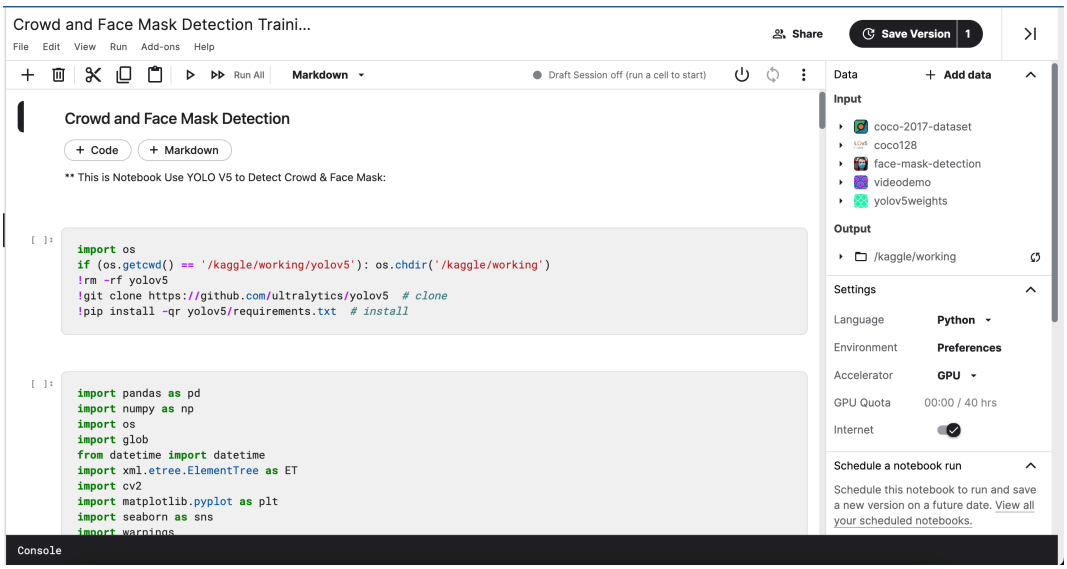


Gambar 2.12 Tampilan Visual Studio Code

2.8.2 Kaggle Notebook

Kaggle Notebook adalah sebuah lingkungan komputasi awan yang memungkinkan analisis yang dapat direproduksi dan kolaboratif. Tipe notebook di Kaggle memiliki 2 tipe, yaitu *Scripts* dan *Notebook*. Tipe *scripts* adalah tipe yang mengeksekusi kode secara sekuensial sedangkan *notebook* adalah tipe yang menggunakan Jupyter *notebook* yang mengeksekusi kode per *cell*, dimana tiap *cell* dapat ditaruh teks biasa atau bahasa pemrograman yang dipilih.

Kaggle Notebook dapat ditulis menggunakan R atau Python. Kaggle Notebook sangat membantu para developer/data scientist/yang lainnya untuk mencari tau dan menjalankan kode-kode *machine learning* atau kode-kode yang membutuhkan spesifikasi mesin yang tinggi. Tampilan dari Kaggle Notebook dapat dilihat pada *Gambar 2.13*.



Gambar 2.13 Tampilan Kaggle Notebook