

BAB II

LANDASAN TEORI

2.1. Teori Terkait

2.1.1. Kadar Air

Kadar air merupakan banyaknya air yang terkandung dalam suatu bahan dan dinyatakan dalam persen (Aventi, 2015). Pentingnya mengetahui kadar air pada suatu bahan pangan karena air dapat mempengaruhi tekstur, penampakan, dan rasa pada bahan pangan tersebut. Kadar air dalam bahan pangan juga menentukan tingkat kesegaran dan keawetannya. Kepentingan yang lain adalah bahwa kadar air diperlukan untuk memenuhi standar komposisi dan peraturan pangan (Aventi, 2015). Penentuan kadar air yang akurat bergantung pada struktur dan komposisinya.

Menurut derajat keterikatan air, kadar air dalam pangan dapat dibagi menjadi empat tipe (Ahmad, 2014), yaitu:

1. Tipe I adalah tipe molekul air yang terikat dengan molekul-molekul air melalui suatu ikatan hidrogen yang berenergi besar.
2. Tipe II adalah tipe molekul air yang membentuk ikatan hidrogen dengan molekul air lain, namun sifatnya berbeda dengan air murni.

3. Tipe III adalah tipe molekul air yang secara fisik terikat dalam jaringan matriks bahan seperti membran, kapiler, serat, dan lain-lain.
4. Tipe IV adalah tipe air yang tidak terikat dalam jaringan suatu bahan atau air murni.

2.1.2. Buah

Buah merupakan sumber vitamin dan mineral yang memiliki pengaruh penting bagi tubuh manusia. Mengonsumsi buah secara rutin dapat mengurangi risiko seseorang terkena penyakit jantung, kanker, peradangan, diabetes, dan lain sebagainya (Noia, 2014). Sebuah studi pada tahun 2014 menyatakan, buah dapat menjadi “pembangkit tenaga listrik” bagi manusia karena kepadatan nutrisi yang tinggi dan kalori yang rendah.

Buah mengandung 80-90% air yang apabila dikonsumsi secara rutin dapat membantu dalam menjaga jumlah cairan dalam tubuh manusia (Pardede, 2014). Setiap buah memiliki kadar air yang berbeda-beda. Buah dengan kadar air yang lebih tinggi dapat menjadi pengganti cairan tubuh dan dapat memberikan energi tambahan bagi tubuh manusia.

2.1.3. Android

Android merupakan sekumpulan perangkat lunak yang digunakan sebagai sistem operasi berbasis Linux bagi telepon seluler seperti *smartphone* dan *computer tablet* yang platformnya terbuka bagi para *developer* untuk menciptakan dan mengembangkan aplikasi yang akan digunakan untuk berbagai macam kebutuhan. Awal mulanya, Google Inc.

membeli Android Inc. yang saat itu merupakan pendatang baru dalam dunia perangkat lunak. Kemudian dalam pengembangan Android, dibentuklah *Open Handset Alliance*, konsorium dari 34 perusahaan *hardware, software*, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia (Ferga Prasetyo, Mubarak, & Subagja, 2017).

2.1.4. *Object Detection*

Object detection atau mendeteksi suatu objek merupakan sebuah metode yang dilakukan dengan membingkai atau *framing* objek yang akan dideteksi dan memisahkan spasial pada *bounding boxes* dan *probabilities* untuk mengenali atau mendeteksi objek dan mencocokkannya dengan data yang tersimpan di *database*.



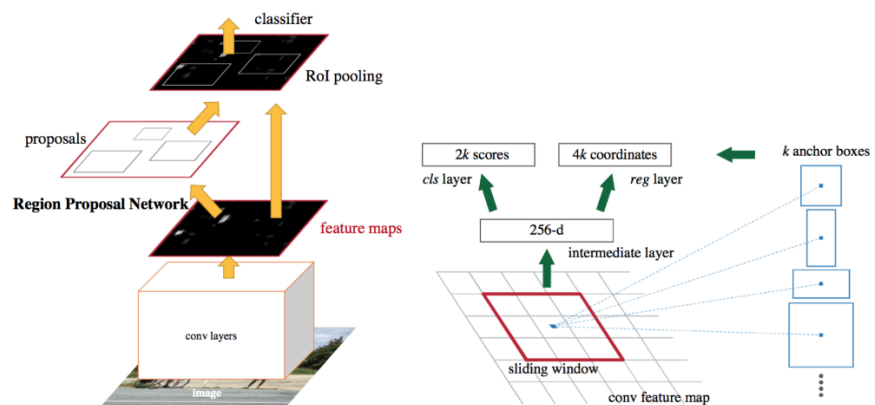
Gambar 2.1. *Object Detection*

Metode ini dilakukan dengan menggunakan *single neural network* untuk memprediksi *bounding boxes* dan *class probabilities* dari seluruh gambar dalam satu kali evaluasi (Redmon, Divvala, Girshick, & Farhadi, 2015). Biasanya algoritma ini memiliki toleransi sebagai pendeteksi sehingga gambar yang hampir mirip akan langsung disamakan. *Object*

detection pada umumnya digunakan untuk mendeteksi wajah, pejalan kaki, atau kendaraan dalam sebuah gambar.

2.1.5. *Faster R-CNN*

Faster R-CNN merupakan salah satu metode *deep learning* yang digunakan untuk mengenali atau mengidentifikasi suatu objek pada gambar (Alamsyah, Informatika, Informasi, & Mdp, 2019). Pengenalan dilakukan dengan menelusuri ciri-ciri yang dimiliki oleh objek pada gambar melalui sejumlah *layer* dengan proses konvolusi atau *Convolutional Neural Network* (CNN).



Gambar 2.2. *Faster R-CNN*

Yang membedakan *Faster R-CNN* dengan metode lainnya adalah *Faster R-CNN* memunculkan *Region Proposal Network* (RPN) yang mengambil gambar (dalam ukuran apapun) sebagai *input*, kemudian mengidentifikasi objek dengan memprediksi *region proposals*. Hasil dari prediksi tersebut kemudian dibentuk kembali menggunakan *RoI Pooling Layer* dan menghasilkan informasi tentang objek dan skornya dalam sebuah kotak pembatas sebagai *output*.

2.1.6. TensorFlow

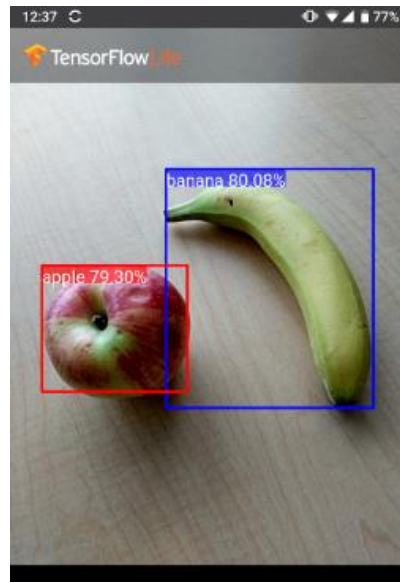
TensorFlow merupakan sebuah *framework* yang digunakan untuk mengeksekusi algoritma *machine learning* dan menjalankan perintah dengan menggunakan data yang dimiliki oleh objek atau target tersebut serta dapat membedakan objek yang satu dengan lainnya (Nurfita & Ariyanto, 2018). Data yang digunakan untuk mengenali objek diperoleh dari gambar yang banyak dan kompleks. *TensorFlow* sangat fleksibel dan dapat digunakan untuk mengekspresikan berbagai macam algoritma. *TensorFlow* memiliki fitur untuk melakukan pelatihan model *Graphic Processing Unit* (GPU) yang memiliki waktu *training* lebih cepat dibandingkan dengan menggunakan *Central Processing Unit* (CPU) (Mureşan & Oltean, 2018) dan memungkinkan pemetaan secara komputasi ke beberapa mesin.

2.1.6.1. TensorFlow Lite

TensorFlow Lite merupakan pengembangan dari *library machine learning* yang dirancang khusus untuk perangkat *mobile* (Alsing, 2018). *TensorFlow Lite* memungkinkan mesin untuk “belajar” di perangkat yang tingkat latensinya rendah dan ukuran *binary*-nya kecil. Untuk menggunakan *TensorFlow Lite* pada perangkat *mobile*, pengguna harus menyediakan Model dalam format *.tflite* yang sudah terlatih atau sudah “belajar” dari beberapa contoh gambar atau kasus yang diberikan.

2.1.6.2. *TensorFlow Lite Object Detection*

TensorFlow Lite Object Detection merupakan bagian dari *TensorFlow Lite* yang melakukan deteksi objek pada gambar dan memberikan informasi mengenai objek tersebut beserta dengan kotak yang membatasi letak objek tersebut dalam gambar (TensorFlow, 2020). Sebuah model *object detection* dilatih untuk mendeteksi keberadaan dan lokasi beberapa kelas objek. Ketika pengguna memberikan gambar ke model, model akan menampilkan label nama objek yang terdeteksi, lokasi kotak pembatas yang berisi setiap objek yang terdeteksi, dan skor yang menunjukkan persentase atau tingkat akurasi bahwa deteksi itu benar.



Gambar 2.3. *TensorFlow Lite Object Detection*

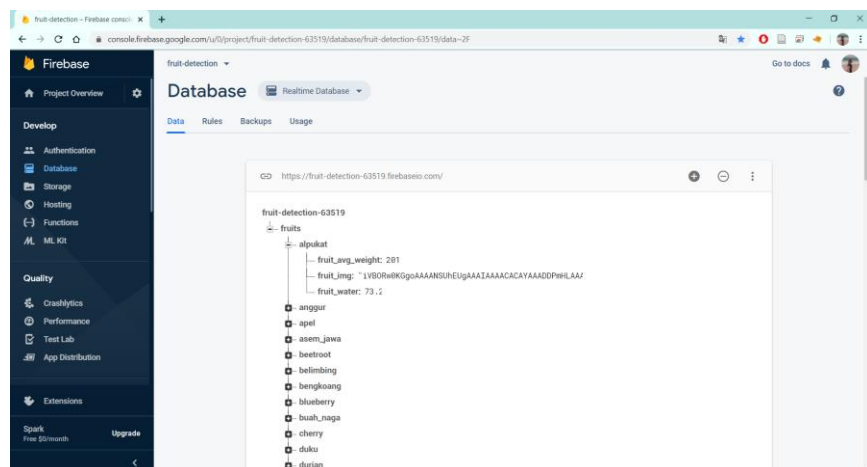
2.1.7. *Firestore*

Firestore merupakan sebuah layanan infrastruktur *backend-as-a-service* (BaaS) yang diakuisisi oleh Google pada Oktober 2014. *Firestore*

memiliki kelebihan yaitu memberikan kemudahan pada para pengembang aplikasi dengan seluruh fitur komplementernya (Justicia, 2017). Hingga saat ini, *Firebase* memiliki fitur *Analytics*, *Cloud Messaging*, *Authentication*, *Realtime Database*, *Cloud Database*, *Hosting*, *Test Lab*, *Crash Reporting*, *Notification*, *Remote Configuration*, *Application Indexing*, *Dynamic Links*, *Invites*, *AdWords*, dan *AdMob*. Seluruh fitur yang dimiliki oleh *Firebase* dikemas dalam sebuah SDK *Firebase* sehingga para pengembang aplikasi maupun *software* dapat membangun infrastruktur tanpa menghabiskan banyak waktu.

2.1.7.1. *Firebase Realtime Database*

Firebase Realtime Database merupakan salah satu fitur yang dimiliki oleh *Firebase* SDK, dimana fitur ini merupakan layanan NoSQL *cloud-hosted database*. Fitur ini menawarkan penyimpanan data yang dapat disinkronisasikan secara *realtime* terhadap seluruh *client* yang terhubung (Justicia, 2017).



Gambar 2.4. *Firebase Realtime Database*

Tiga kemampuan inti yang dimiliki oleh *Firestore* adalah *realtime*, *offline*, dan dapat diakses melalui perangkat *client*. Aplikasi yang menggunakan *Firestore* akan tetap responsif bahkan saat dalam keadaan *offline* karena *Firestore* SDK dapat mempertahankan data dan perubahannya dalam media penyimpanan *client*. Ketika *client* terhubung dengan jaringan internet, maka *Firestore* SDK akan melakukan penyesuaian otomatis (sinkronisasi) atas perubahan data yang disimpan dalam media penyimpanan *client* dengan kondisi terbaru dari server *Firestore*.

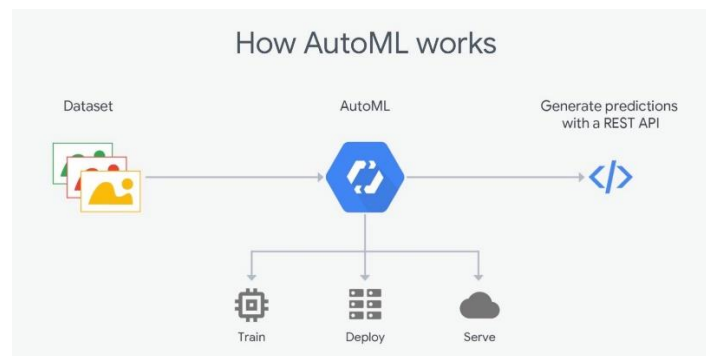
2.1.8. Google Cloud Platform

Google Cloud Platform merupakan layanan *cloud computing* yang ditawarkan oleh Google. *Google Cloud Platform* menawarkan solusi untuk kebutuhan penyimpanan, analisis, *big data*, dan *machine learning* bagi pengembangan aplikasi atau sistem (Google, n.d.-b). Dengan *training* dan *resource* yang disediakan oleh Google, pengguna dapat mempercepat pembuatan aplikasi. Pengguna juga dapat memanfaatkan fitur-fitur *cloud* yang canggih dan sangat membantu dalam *big data* dan *machine learning* untuk menciptakan produk yang lebih baik dan aplikasi yang memiliki fungsional tinggi.

2.1.8.1. AutoML Vision

AutoML Vision merupakan salah satu layanan dalam *Google Cloud Platform* yang memanfaatkan serangkaian produk

machine learning sehingga memungkinkan *developer* untuk melatih model khusus sesuai kebutuhan dengan cepat dan mudah (Google, n.d.).

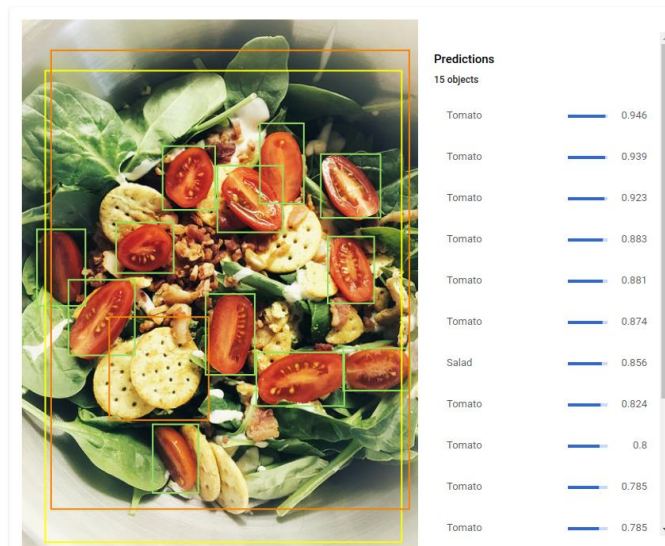


Gambar 2.5. *AutoML Vision*

AutoML Vision akan melatih dataset yang sudah dibuat dan menghasilkan model yang dapat di-*export* dan digunakan sesuai dengan kebutuhan.

2.1.8.2. *AutoML Vision Object Detection*

AutoML Vision Object Detection merupakan bagian dari fitur *AutoML Vision* yang membantu *developer* dalam mengembangkan aplikasi dengan melatih *custom machine learning model* yang mendeteksi objek secara individual dalam gambar tertentu beserta dengan kotak dan labelnya (Google, n.d.-a). Fitur ini dapat mendeteksi objek dalam suatu gambar dan memberikan informasi tentang objek dan dimana objek tersebut ditemukan dalam gambar. Model yang sudah di-*training* dengan *AutoML Vision Object Detection* dapat di-*export* dalam bentuk *TensorFlow Lite* dan digunakan untuk perangkat *mobile*.



Gambar 2.6. AutoML Vision Object Detection

2.1.9. Android Studio

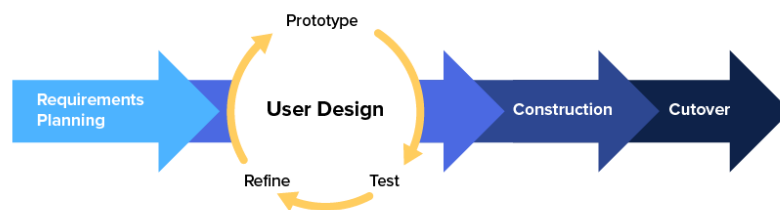
Android Studio merupakan sebuah *Integrated Development Environment* (IDE) khusus untuk membangun aplikasi yang berjalan pada platform Android (Buwono et al., 2017). Bahasa pemrograman utama yang digunakan adalah Java, sedangkan untuk membuat tampilan antarmuka menggunakan bahasa XML. Android Studio terintegrasi dengan Android *Software Development Kit* (SDK) untuk men-*deploy* aplikasi ke perangkat Android. Setiap proyek yang dikerjakan di Android Studio berisi satu atau lebih model dengan *source code* dan *resources* yang mendukung perancangan aplikasi.

2.1.10. Rapid Application Development

Rapid Application Development merupakan metode pengembangan yang melakukan pengembangan jauh lebih cepat dan menghasilkan kualitas yang lebih baik dibandingkan dengan hasil yang dicapai melalui model lain. Pengembangan sistem menggunakan metode RAD dapat dilakukan dalam

waktu yang relatif singkat karena menggunakan pendekatan konstruksi berorientasi objek. Dengan menggunakan metode RAD, biaya yang dikeluarkan pengembangan relatif lebih kecil dan memberikan pengguna untuk memeriksa dan memberikan *feedback* terhadap sistem secara cepat.

Metode *Rapid Application Development* (RAD) memiliki 4 (empat) tahapan dasar sebagai berikut (Rosenblatt & Shelly, 2012).



Gambar 2.7. Tahapan Metode *Rapid Application Development*

1. *Requirements Planning*

Setiap pengembangan sistem memerlukan perencanaan akan apa yang dibutuhkan oleh pengguna. Pada tahap ini akan dibuat ruang lingkup penelitian, tujuan dari penelitian, daftar kebutuhan sistem, dan daftar kebutuhan *user*. Tahap ini akan berakhir ketika seluruh kebutuhan telah teridentifikasi secara lengkap dengan lancar (Rosenblatt & Shelly, 2012).

2. *User Design*

Pengembangan sistem berlanjut dengan membuat prototipe dengan berbagai fitur dan fungsi yang akan merepresentasikan alur dari sistem. Prototipe memungkinkan *user* untuk melakukan komunikasi atau perubahan terhadap sistem yang akan dibangun

hingga *user* menyetujui sistem yang dibutuhkan (Rosenblatt & Shelly, 2012).

3. *Construction*

Sistem akan mulai dibangun dan dikembangkan sesuai dengan kebutuhan *user*. *User* akan berpartisipasi dengan memberikan *feedback* agar segera dilakukan perbaikan untuk kemudian dilakukan uji coba kepada *user* (Rosenblatt & Shelly, 2012).

4. *Cutover*

Tahap terakhir adalah proses *testing* dan perbaikan sebelum diberikan kepada *user*. Perbaikan sistem akan dilakukan setelah menerima *feedback* dari *user* yang telah mencoba sistem (Rosenblatt & Shelly, 2012).

2.1.11. Mean Opinion Score

Mean Opinion Score atau MOS adalah penilaian subjektif oleh responden terhadap rata-rata nilai pada skala yang telah ditetapkan untuk menilai kinerja atau fungsi suatu sistem digital (Streijl, Winkler, & Hands, 2016). MOS digunakan dalam menghitung hasil pengujian atau *survey* dari pengujian dimana nilai rata-rata suatu indikator yang digunakan antara 1 sampai 5, dimana 1 berarti buruk atau sangat tidak setuju dan 5 adalah yang paling baik atau sangat setuju. Penilaian ini dilakukan berdasarkan pengamatan atau pengujian yang dilakukan manusia sehingga hasil baik maupun buruknya bergantung pada penilaian masing-masing responden.

Berikut Tabel 2.1. menjabarkan bobot penilaian yang digunakan dalam *Mean Opinion Score* (MOS).

Tabel 2.1. Mean Opinion Score

MOS	Keterangan	Bobot Nilai
SS	Sangat Setuju	5
S	Setuju	4
N	Netral	3
TS	Tidak Setuju	2
STS	Sangat Tidak Setuju	1

2.2. Penelitian Terdahulu

Tabel 2.2 Penelitian Terdahulu

Jurnal 1	
Nama Jurnal	TRITA-EECS-EX, 335 (2018)
Judul Penelitian	<i>Mobile Object Detection using TensorFlow Lite and Transfer Learning</i>
Peneliti	Oscar Alsing
Hasil Penelitian	Hasilnya menunjukkan ada peningkatan dalam kinerja mendeteksi objek dibandingkan dengan algoritma yang ada dengan menggunakan <i>transfer learning</i> yang diadaptasi untuk penggunaan mobile.
Kesimpulan	Jurnal ini menganalisa kelayakan model yang digunakan untuk mendeteksi objek dalam <i>realtime video feeds</i> pada perangkat seluler.
Jurnal 2	
Nama Jurnal	Buletin Penelitian Kesehatan, Vol. 44, 3 (2016), p. 4-10
Judul Penelitian	Gambaran Konsumsi Sayur dan Buah Penduduk Indonesia dalam Konteks Gizi Seimbang: Analisis Lanjut Survei Konsumsi Makanan Individu (SKMI) 2014

Peneliti	Hermina dan Prihatini S
Hasil Penelitian	Untuk mencapai pola makan sehat dengan gizi seimbang diperlukan konsumsi sayur dan buah dengan proporsi yang sesuai dengan umur, baik penduduk yang tinggal di perkotaan maupun di pedesaan.
Kesimpulan	Konsumsi sayur dan buah oleh penduduk Indonesia masih tergolong rendah terutama golongan remaja sehingga diperlukan analisis yang memberikan informasi tentang proporsi yang sesuai untuk mencapai pola makan sehat.
Jurnal 3	
Nama Jurnal	Acta Univ. Sapientiae, Informatica, Vol. 10, 1 (2018), p. 26-42
Judul Penelitian	<i>Fruit Recognition from Images using Deep Learning</i>
Peneliti	Horea Muresan, Mihai Oltean
Hasil Penelitian	Gambar buah-buahan yang terdapat dalam dataset dapat diklasifikasi menggunakan <i>TensorFlow library</i> .
Kesimpulan	Penelitian yang ditujukan untuk meningkatkan akurasi pada <i>neural network</i> dan memperluas dataset dengan memperbanyak jenis buah-buahan.

Berdasarkan beberapa jurnal yang dijadikan sebagai referensi dapat diambil keterkaitan yang digunakan dalam penelitian, yaitu dalam jurnal (Alsing, 2018) penggunaan model *TensorFlow Lite* akan digunakan pada pengembangan sistem, namun objek yang akan digunakan adalah buah-buahan. Dalam jurnal (Hermina & S, 2016), analisis yang dilakukan akan memberikan informasi tentang jumlah kadar air dalam buah yang dikonsumsi agar penduduk dapat mencapai pola hidup sehat dengan cairan tubuh yang cukup. Dalam jurnal (Mureşan & Oltean, 2018), korelasi

antara penggunaan *library TensorFlow* dan dataset gambar buah-buahan akan digunakan dalam melakukan pembuatan model. Namun, dataset yang digunakan merupakan dataset yang berisi foto buah yang diambil secara langsung, bukan foto yang diambil dari internet.