

BAB II

LANDASAN TEORI

2.1 Pelestarian

Pelestarian dalam Kamus Besar Bahasa Indonesia (KBBI) berasal dari kata abadi yang artinya tetap apa adanya, tidak berubah, abadi dan abadi. Kata tahan lama jika dalam bahasa Indonesia ditambah awalan pe dan akhiran -an, maka akan menjadi kata kerja. Kata itu akan menjadi kata konservasi, berarti pelestarian menurut Kamus Besar Bahasa Indonesia yaitu proses, cara, perbuatan pelestarian, perlindungan dari kehancuran atau kerusakan, pengawetan, pengawetan. Konservasi adalah upaya untuk melindungi, mengembangkan, dan memanfaatkan budaya yang hidup (Peraturan Umum Menteri Dalam Negeri dan Menteri Kebudayaan dan Pariwisata tentang Pedoman Pelestarian Budaya Tahun 2009).

Widjaja dalam buku Jacobus mendefinisikan pelestarian sebagai suatu kegiatan atau kegiatan yang dilakukan secara berkesinambungan, berorientasi dan terpadu yang ditujukan untuk mencapai tujuan tertentu yang mencerminkan adanya sesuatu yang tetap dan abadi, dinamis, fleksibel dan selektif.

2.2 Budaya

Budaya adalah konsep dasar dalam antropologi. E.B. Taylor, salah satu divisi antropolog, mendefinisikan budaya sebagai sesuatu yang mencakup semua pengalaman manusia. EB Taylor berpendapat bahwa budaya mencakup pengetahuan, seni, moral, hukum, dan keterampilan serta perilaku lain yang diterima atau dipelajari oleh orang dan anggota masyarakat. Dari konsep tersebut dapat dikatakan bahwa budaya merupakan produk yang dibuat oleh manusia, dan pada saat yang sama budaya juga membentuk manusia dalam kehidupan sehari-hari [8].

Dalam konteks budaya, manusia dipandang sebagai hewan simbolik, makhluk simbolik, dan makhluk budaya yang kehidupannya dibentuk oleh produk budaya. Selanjutnya, budaya diwarisi bukan oleh kode genetik, tetapi melalui proses pengkodean, yaitu proses interaksi manusia di mana seorang individu belajar dan menerima budayanya. Orang secara sadar menginternalisasi budaya mereka melalui pembelajaran langsung dan secara tidak sadar melalui interaksi [8].

2.3 E-Culture

Istilah *e-Culture* mulai berkembang di Eropa, ketika di Belanda pada awalnya *e-Culture* dipersepsikan sebagai penggunaan instrumen TIK dalam ranah budaya. Kemudian Virtueel Platform juga mengadopsi istilah *e-Culture*, misalnya terbitnya tahun 1999 tentang "*E-Culture: Cultural Policy For Innovation*". Pada pertengahan 1990-an digunakan dalam dua konteks, pertama berkaitan dengan pelestarian warisan budaya berbasis digital, dan juga dalam arti yang kontras dengan e-commerce. Dalam perjalanan tahun 1990-an *e-Culture* meningkat digunakan dalam kebijakan pemerintah, dan tidak hanya di Belanda [2].

Istilah *e-Culture* mulai dipopulerkan oleh pemerintah Belanda pada bulan April 2002 dalam rancangan kebijakan Departemen Pendidikan, Kebudayaan dan Ilmu Pengetahuan. Pada tahun 2003 Dewan Kesenian menerbitkan laporan berjudul "*From ICT to e-Culture*". Laporan tersebut menekankan bahwa digitalisasi mempengaruhi seluruh spektrum distribusi produksi budaya. Internet dan media digital berdampak pada cara seniman mengekspresikan diri, bagaimana warisan budaya disajikan, bagaimana perpustakaan menyediakan informasi dan dapat diakses oleh publik, dan bagaimana informasi, debat publik dan budaya diperlakukan di media. *e-Culture* berarti semua proses "ekspresi, refleksi dan berbagi dalam domain digital" [2].

e-Culture juga merupakan sektor yang relatif baru dibandingkan dengan disiplin ilmu atau bidang penelitian lainnya. Implementasi *e-Culture* adalah sektor tersendiri, dan bagian dari rantai nilai budaya UNESCO. Jika potensi kekayaan budaya dapat dimanfaatkan semaksimal mungkin dengan teknologi informasi dan komunikasi hal ini akan mendorong peningkatan kesejahteraan masyarakat Indonesia karena kita memiliki kekayaan budaya yang lebih banyak dari negara lain. Menurut survei Badan Pusat Statistik, kekayaan budaya Indonesia mencakup lebih dari 300 suku bangsa atau etnis di Indonesia, atau tepatnya 1.340 suku bangsa [2].

Menurut Dirk dan Esmans, *E-Culture* mengacu pada hubungan antara teknologi informasi dan komunikasi (TIK) dan media digital, mengacu pada produksi dan konsumsi seni dan budaya. Dengan munculnya media digital di sektor seni dan budaya, bentuk seni baru dan kemungkinan baru diciptakan untuk produksi, presentasi, dan penyimpanan produk seni dan budaya. ICT mengubah cara menciptakan budaya, menyebarkan, melestarikan, dan keterlibatan. Pemahaman

ini menunjukkan bahwa terdapat dua hal dalam definisi *E-Culture*, yaitu cara hidup orang berbasis teknologi digital dan mendigitalisasikan atau elektronik faktor budaya [2].

2.4 Bogor

Kota Bogor adalah sebuah kota di provinsi Jawa Barat, Indonesia. Kota ini terletak 59 km sebelah selatan ibukota Jakarta dan wilayahnya berada di tengah wilayah Kabupaten Bogor. Sebelumnya luas wilayahnya 21,56 km², namun sekarang telah meningkat menjadi 118,50 km² dan berpenduduk 1.030.720 jiwa (2014). Bogor dikenal sebagai kota hujan, karena memiliki curah hujan yang sangat tinggi. Kota Bogor terdiri dari 6 kecamatan yang terbagi menjadi 68 kecamatan. Pada masa penjajahan Belanda, Bogor dikenal sebagai *Buitenzorg* (diucapkan: *boit'nzôrkh*", *boeit*') yang berarti "tanpa kecemasan" atau "aman tenteram" [9].

Bogor (berarti "enau") telah lama dikenal dijadikan pusat pendidikan dan penelitian pertanian nasional. Di sinilah berbagai lembaga dan balai penelitian pertanian dan biologi berdiri sejak abad ke-19. Salah satunya yaitu, Institut Pertanian Bogor, berdiri sejak awal abad ke-20 [9].

Kota Bogor memiliki sejarah panjang dalam pemerintahan mengingat sejak zaman Kerajaan Pajajaran menurut bukti-bukti yang ada dari prasasti, nama desa itu disebut Lawangtung, Lawang Saketeng, Jerokuta, Baranangsiang dan Leuwi Sipatahunan yang meyakini bahwa Pakuan adalah Ibukota Pajajaran terletak di kota Bogor. Pakuan sebagai Pusat Pemerintahan Pajajaran terkenal pada pemerintahan Prabu Siliwangi (Sri Baginda Maharaja), penobatan berlangsung pada tanggal 3 Juni 1482 yang kemudian menjadi hari lahir Bogor karena sejak tahun 1973 ditetapkan oleh Pemerintah Kota. Pemerintah dan DPRD sebagai hari jadi Kota Bogor dan masih diperingati setiap tahun hingga saat ini. Setelah invasi tentara Banten ke Pakuan Pajajaran, catatan tentang kota Pakuan hilang, baru terungkap lagi setelah kedatangan kelompok ekspedisi Belanda yang dipimpin oleh Scipio dan Riebeck pada tahun 1687, dan mereka memeriksa Batutulis. Prasasti dan situs lain menunjukkan bahwa Bogor adalah pusat pemerintahan Pajajaran Pakuan. Pada tahun 1745 Gubernur Jendral Hindia Belanda pada waktu itu bernama Baron Van Inhoff membangun Istana Bogor, seiring dengan pembangunan jalan Raya Daenless yang menghubungkan Batavia dengan Bogor, sehingga keadaan Bogor mulai berkembang [9].

Kota bogor memiliki visi berupa “Kota jasa yang nyaman dengan masyarakat madani dan pemerintahan amanah”, serta misi yaitu:

1. Mengembangkan perekonomian masyarakat dengan titik berat pada jasa yang mengoptimalkan pemanfaatan sumber daya yang ada.
2. Mewujudkan kota yang bersih, indah, tertib, dan aman dengan sarana dan prasarana perkotaan yang memadai dan berwawasan lingkungan.
3. Meningkatkan kualitas sumber daya manusia yang beriman dan berketerampilan.
4. Mewujudkan pemerintahan kota yang efektif dan efisien serta menjunjung tinggi supremasi hukum.

2.4.1 Peninggalan Sejarah Kota Bogor Sebagai Kekuatan Lokal

Bogor adalah salah satu kota pedalaman terpenting pada masa kolonial, karena Bogor (dahulu *Buitenzorg*) pernah menjadi ibu kota pemerintah kolonial Belanda di Hindia Timur, sejak Gubernur Jenderal Van Alting (1780). Bogor pada awalnya dikonsepsikan dengan penguasaan dan pertanian di lahan perkebunan yang dikelola oleh tuan tanah, yang akhirnya berkembang setelah dihubungkan dengan jalan raya Pos (*Groote Postweg*) yang dibangun pada tahun 1811 oleh Daendels dan kemudian rel kereta api (*BataviaBuitenzorg*) pada tahun 1873 [10].

Banyak artefak fisik kota yang dibangun selama masa kolonial ini, termasuk Istana Bogor (sebelumnya dikenal sebagai Vila *Buitenzorg*), yang didirikan atas prakarsa Gubernur Jenderal GW Baron van Imhoff. Gedung ini juga pernah menjadi kantor resmi Gubernur Jenderal VOC dan Gubernur Jenderal Hindia Belanda hingga tahun akhirnya menjadi Istana Kepresidenan Republik Indonesia. Halaman Istana Buitenzorg dibangun di Botanical Garden (Kebun Raya) oleh seorang ahli botani Jerman, Profesor R.C. Kebangkitan dan dibuka sebagai Kebun Raya Bogor pada tahun 1887. Dengan luas 87 hektar, Kebun Raya Bogor saat ini merupakan kebun raya terbesar di Asia Tenggara dan merupakan artefak alam yang khas bagi kota Bogor [10].



Gambar 2.1 Tugu Kujang

Sumber: <https://sentul.city/tugu-kujang-bogor-ciri-khas-bogor/>



Gambar 2.2 Kebun Raya Bogor

Sumber: <https://bogor.ayoinonesia.com/berita-bogor/pr-31872210/Mulai-7-Juli-Kebun-Raya-Bogor-Buka-Kembali>

Secara fisik, Kota Bogor memiliki banyak peninggalan sejarah yang berkontribusi terhadap identitas kota. Namun karena fenomena perkotaan yang merajalela saat ini, karakter historis Kota

Bogor mulai terdesak oleh dampak pembangunan wilayah. Bogor kini menjadi kota yang merepresentasikan perpaduan budaya tradisional Sunda, budaya kolonial, dan urbanisme modern [10].

Persaingan antara pemerintah daerah terus berlanjut di kota. Oleh karena itu, saat ini terdapat 3 jenis kawasan yang menentukan karakter kota Bogor, yaitu Zona Sejarah Kolonial (kekuatan lokal), Zona Pengembangan Ekonomi Internal (penggabungan kekuatan lokal dan regional) dan Kawasan Pemukiman Terkait Komuter pengembangan struktur (kekuatan regional). Sayangnya, ketiga karakter dalam perkembangan kota tersebut tidak mampu bersatu padu menjadikan Bogor kota yang penuh identitas. Identitas kota (*urban identity*) dapat diekspresikan melalui faktor-faktor yang menyebabkan keterikatan emosional (keterikatan dengan tempat tinggal) penduduk kota. Meskipun Kawasan Bersejarah Kolonial saat ini merupakan kawasan Bogor yang paling kurang berkembang, namun kawasan dengan konsep taman kota memberikan kontribusi paling besar dalam hal tingkat keterikatan dengan tempat tinggal masyarakat Bogor [10].

2.4.2 Sosial Budaya Daerah Bogor

Bogor adalah kota yang terletak di sebelah Jakarta. Bogor memiliki banyak budaya yang berbeda, terutama karena Bogor merupakan daerah yang strategis dan memiliki banyak tempat wisata. Kota Bogor merupakan salah satu wilayah Tatar Sunda yang kaya akan sumber daya arkeologi dari berbagai periode budaya. Monumen-monumen tersebut antara lain berasal dari masa prasejarah hingga masa kebudayaan selanjutnya, seperti masa klasik, masa Islam, serta masa pengaruh Eropa [11].

Hampir semua wilayah kota Bogor memiliki tempat wisata dan tempat untuk menemukan monumen ini hampir tersebar di seluruh wilayah Bogor. Dari masa prasejarah, tinggalan arkeologi sementara mungkin telah menjadi bagian dari masa prasejarah wilayah Kota Bogor, termasuk peninggalan dari tradisi budaya yang dimulai dan berkembang dari masa pertanian (Neolitik), terutama berupa benda dan bangunan yang saling berhubungan. tradisi atau ritual yang berkaitan dengan penghormatan terhadap arwah dan leluhur, khususnya benda dan bangunan yang termasuk dalam kelompok tradisi budaya megalitik. Sebelumnya, menurut von Heine Geldern (1945)

menyimpulkan bahwa tradisi budaya ini diperkenalkan dan dikembangkan di Nusantara sejak 3000 SM, kemudian terus berkembang seiring berjalannya waktu sejarah. Karena perkembangan budaya yang cukup lama inilah yang kemudian disebut dengan tradisi budaya megalitik [11].

2.5 Image Recognition

Image Recognition adalah proses mengidentifikasi dan mengenali objek atau fitur dalam gambar dan video digital. konsep ini digunakan untuk melakukan berbagai tugas visual berbasis mesin. Menambahkan meta-tag ke konten gambar, melakukan pencarian konten gambar, dan mengendarai robot otonom, mobil self-driving, dan sistem penghindaran kecelakaan. Otak manusia dan hewan dapat dengan mudah mengenali objek, tetapi komputer berjuang dengan tugas ini. Perangkat lunak pengenalan gambar membutuhkan pembelajaran mesin yang mendalam. Kinerja prosesor jaringan saraf convolutional adalah yang terbaik. Jika tidak, tugas-tugas tertentu membutuhkan komputasi yang intensif dan membutuhkan banyak daya. Algoritma pengenalan gambar bekerja berdasarkan model 3D komparatif dan dapat dilihat menggunakan deteksi tepi atau dari sudut yang berbeda berdasarkan komponen demi komponen. Algoritme pengenalan gambar sering dilatih dalam pembelajaran komputer terpandu dari jutaan gambar yang telah diberi label sebelumnya. Algoritma pengenalan gambar yang umum adalah pengenalan karakter optik, pencocokan pola dan gradien, pencocokan plat nomor kendaraan, serta identifikasi adegan atau deteksi perubahan adegan.

Image Recognition merupakan bagian dari kecerdasan buatan (AI) yang bertujuan untuk meningkatkan data citra. Dengan kata lain, ini bertujuan untuk mempersiapkan gambar yang akan diambil agar dapat dikerjakan dan lebih mudah untuk diproses pada langkah selanjutnya [4]. Langkah-langkah utama dalam proses *Image Recognition* adalah:

1. Peningkatan Kualitas Citra

Penyempurnaan citra dapat dilakukan dengan mengubah warna citra yang semula berwarna menjadi abu-abu atau yang dikenal dengan proses grayscale [4].

2. Segmentasi Citra

Segmentasi citra dimaksudkan untuk menyeleksi dan memisahkan suatu objek dari keseluruhan citra. Pada tahap ini, terdapat metode downampling untuk mengurangi jumlah piksel dan untuk menghilangkan sebagian informasi citra. Selain itu, pada tahap segmentasi gambar, menghilangkan noise atau derau dari dalam gambar, seperti kurangnya keburaman dari pada gambar [4].

3. Pengenalan dan Interpretasi

Identifikasi dan interpretasi Pada tahap ini gambar yang diproses akan direpresentasikan dalam bentuk yang sesuai untuk kualitas yang baik. Interpretasi juga mencakup penekanan pada makna objek yang dikenali [4].

2.6 API

Application programming interface (API) adalah konsep fungsionalitas antarmuka pemrograman aplikasi, yang merupakan cara aplikasi untuk diakses dan digunakan oleh pihak lain tanpa mengubah struktur kode utama atau kerangka kerja yang mendasarinya. serta memfasilitasi komunikasi antar sistem meskipun berbeda platform (Wijonarko & Mulya, 2018). *Web Service* adalah API yang menyediakan akses pengguna selama pengambilan data. Berkat struktur *Representative State Transition* (ReST) yang dioperasikan melalui *Hypertext Transfer Protocol* (HTTP), ini berisi file *Javascript Object Notation* (JSON), yang akan ditampilkan kepada pengguna saat mengakses API [12].

API memiliki beberapa jenis seperti *Public API* atau biasa disebut *Open API* merupakan jenis API yang dapat digunakan siapa saja dan merupakan jenis API paling mudah digunakan. Tidak seperti *Public API*, *Private API* adalah jenis API yang tidak dapat digunakan secara publik. Jenis API ini biasanya dibuat untuk tujuan internal saat mengembangkan aplikasi tertentu. *Partner API* dapat digunakan untuk tujuan publik, tetapi hanya jika sudah memiliki izin untuk menggunakannya. Seperti halnya API publik, harus terlebih dahulu mendaftar ke penyedia API tersebut kemudian gunakan hanya di aplikasi tertentu sesuai kontrak. *Composite API* merupakan API yang menyimpan data dari berbagai server atau hosting dalam satu tempat.

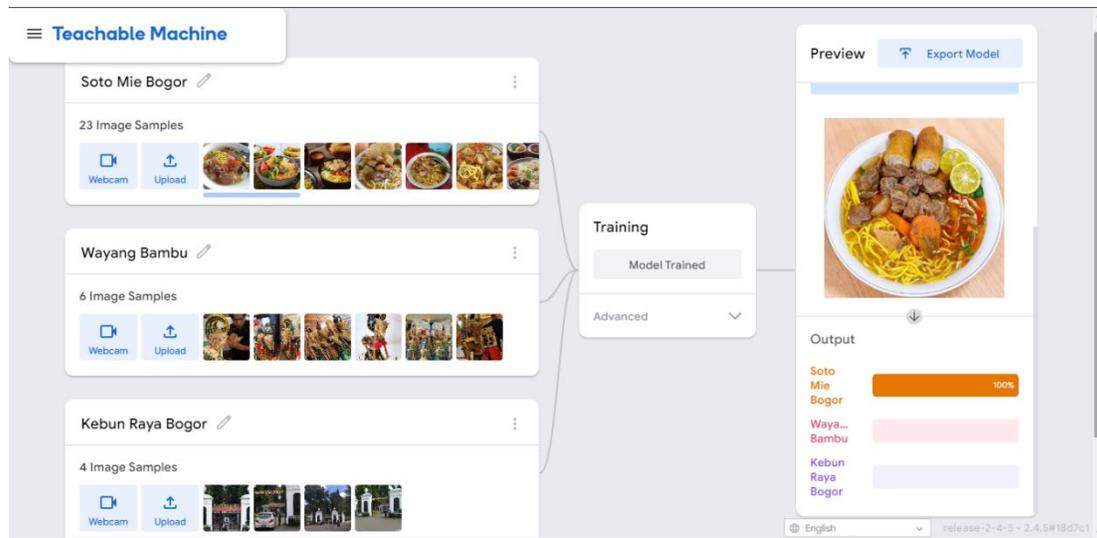
API memiliki beberapa manfaat seperti mempermudah membangun aplikasi yang fungsional, API memudahkan pembuatan aplikasi yang fungsional dan kompleks. Aplikasi yang

dikembangkan memiliki fungsionalitas aplikasi target tanpa perlu menambahkan data secara manual. Pengembangan aplikasi menjadi lebih efisien, Dengan API, tidak perlu berkomunikasi langsung dengan aplikasi lain yang ingin disambungkan cukup melalui API. Ini sangat berguna saat membangun aplikasi lintas platform menggunakan beberapa layanan secara bersamaan. Meringankan beban server, Dengan menggunakan API, tidak perlu menyimpan semua data yang dibutuhkan di server itu sendiri, yang perlu dilakukan adalah meminta API untuk mendapatkan data terbaru dari server aplikasi asli. Status ini tidak membebani server. Pada akhirnya, risiko situs web menjadi tidak tersedia karena server mati berkurang.

API memiliki tiga arsitektur yang developer aplikasi sering gunakan seperti RPC (*Remote procedure call*) merupakan teknologi pembuat komunikasi *client side* dan *server side* dengan konsep sederhana. SOAP (*Simple Object Access Protocol*) merupakan arsitektur menggunakan XML (*Extensible Markup Language*) dimana semua data disimpan dalam dokumen. REST (*Representational State Transfer*) menggunakan JSON sebagai bentuk data yang membuatnya lebih ringan.

2.7 Teachable machine

Teachable machine adalah antarmuka berbasis web yang memungkinkan orang untuk melatih model klasifikasi *machine learning* mereka sendiri, tanpa coding, menggunakan webcam, gambar, atau suara mereka. Menggunakan pembelajaran transfer, teknik *machine learning*, untuk menemukan pola dan tren dalam gambar atau sampel suara, dan membuat model klasifikasi sederhana dan mudah dalam hitungan detik. Dengan pembelajaran transfer, pengguna dapat menambahkan sendiri data dan melatih kembali model di atas yang sebelumnya dilatih model dasar yang telah mempelajari domain tertentu dari kumpulan data besar. Misalnya, model dasar (mobilenet) untuk bagian gambar dari *Teachable machine* awalnya dilatih untuk mengenali 1000 kelas (seperti anjing, telepon, tempat tidur, trombon, dll). Sifat dasar yang digunakan mobilenet untuk mengenali kelas-kelas itu dapat digunakan untuk mengidentifikasi yang baru kelas yang didefinisikan pengguna. Kompleksitas ini tersembunyi dari pengguna, yang hanya diuntungkan dengan membutuhkan lebih sedikit data dan waktu pelatihan untuk membuat model yang berguna dan akurat [13]. Berikut merupakan gambaran pelatihan model pada *Teachable machine*.



Gambar 2.3 Tampilan Teachable machine

Teachable machine merancang mesin yang dapat diajar untuk digunakan oleh para pendidik untuk mengajarkan konsep *machine learning* dengan mudah, untuk prototipe, desainer & peneliti untuk menguji konsep interaktif dengan cepat dengan *machine learning*, dan bagi pengguna penyandang disabilitas untuk membuat personal model yang dapat mereka kendalikan. Sejak diluncurkan, orang-orang telah melatih lebih dari 125.000 model klasifikasi *machine learning* dengan *Teachable machine* [13].

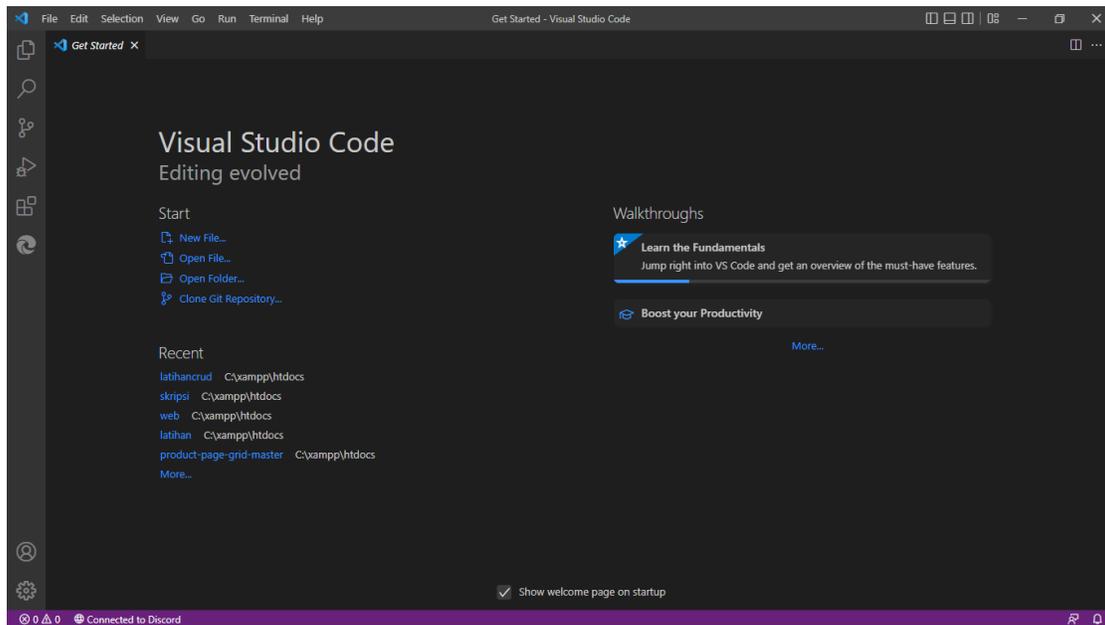
Kontribusi utama proyek *Teachable machine* yaitu membuat:

1. Antarmuka masukan yang fleksibel dan umum untuk pembuatan Model klasifikasi *machine learning* yang dapat dipelajari dengan mudah dan digunakan tanpa pengalaman atau keahlian sebelumnya dalam *machine learning* atau pengkodean.
2. Serangkaian keputusan produk yang memungkinkan pembelajaran dan eksperimen untuk pengguna baru *machine learning*.
3. Contoh bagaimana konten seputar antarmuka memungkinkan orang untuk mempelajari konsep *machine learning*.

2.8 Visual studio code

Visual studio code adalah editor *source code* yang dikembangkan oleh Microsoft untuk Windows, Linux dan MacOS. Ini termasuk dukungan debugging, pengujian GIT terintegrasi, penyorotan sintaks, penyelesaian *SmartCode*, cuplikan kode, dan pemfaktoran kode. *Visual studio code* memiliki beberapa pengaturan yang dapat disesuaikan, sehingga pengguna dapat mengubah tema editor, pintasan pada *keyboard*, dan preferensi. *Visual studio code* gratis dan *open source*, meskipun unduhan resmi berada di bawah lisensi kepemilikan.

Visual studio code secara langsung mendukung bahasa pemrograman *Javascript*, *Typescript* dan *Node.js*, serta bahasa pemrograman lainnya seperti *Python*, *Java*, *C++*, *C#*, *Go*, dan lainnya dengan bantuan plugin yang dapat dipasang melalui *Marketplace Visual studio code*. Gambar 2.4 merupakan tampilan *Visual studio code*:



Gambar 2.4 Visual studio code

2.9 Internet

Internet adalah sistem jaringan komputer global yang saling terhubung menggunakan protokol internet (TCP/IP) untuk menghubungkan perangkat di komputer di seluruh dunia. Internet adalah jaringan dari banyak jaringan yang terdiri dari jaringan pribadi, umum, akademik, bisnis,

dan pemerintah lokal. Dihubungkan oleh barisan bahasa pemrograman yang luas dan mencakup peralatan elektronik, nirkabel, dan teknologi jaringan optik. Internet memberikan berbagai sumber informasi dan jasa, seperti akses *World Wide Web* (WWW), surat elektronik, telepon, dan jaringan *peer-to-peer* untuk saling berbagi berkas [14].

Pada awalnya Internet hanya terdiri dari beberapa jaringan komputer kecil yang didirikan oleh Departemen Pertahanan Amerika ARPANET untuk tujuan riset, yakni sekitar tahun 1969. Pada tahun 1971 ARPANET baru terdiri dari lima belas titik jaringan (*nodes*) dengan 23 host (server induk), dan aplikasi yang canggih waktu itu adalah elektronik mail. Tahun 1973 ARPANET membentuk WAN yang terhubung dari Amerika ke jaringan di Norwegia dan Inggris. Tahun 1983 ARPANET baru terdiri dari 235 Host. Angka ini melonjak pada tahun 1989 hingga mencapai 100 ribu host. Tahun 1990 ARPANET berganti nama menjadi INTERNET. Pada ulang tahunnya yang ke-25 Internet sudah terdiri dari 2 juta lebih host dan meningkat menjadi 2 kali lipat pada tahun 1995 (4 juta host). Internet bukan lagi sekedar jaringan yang meliputi Amerika dan Eropa, tapi sudah meliputi seluruh bagian dunia, termasuk Indonesia [15].

2.10 Website

Menurut Sugiyanto pada 2013 situs web (*website*) adalah alamat URL (Uniform Resource Locator) yang berfungsi sebagai tempat penyimpanan data dan informasi tentang topik tertentu. Menurut Nazrul pada 2006 halaman web atau web page adalah kumpulan halaman yang menampilkan informasi data tekstual, data gambar diam atau bergerak, data animasi, audio, video dan/atau kombinasi semuanya, baik statis maupun dinamis, membentuk rangkaian bangunan yang saling berkaitan, masing-masing yang terhubung ke jaringan halaman (*hyperlink*). Bersifat statis jika isi informasi *website* tetap, jarang berubah dan isi informasi hanya searah dengan pemilik *website* tersebut. Bersifat dinamis jika konten informasi *website* terus berubah dan konten informasi bersifat interaktif dua arah dari pemilik *website* dan pengguna [16].

Menurut Widodo pada 2013, *website* atau sering disingkat web, adalah halaman teks huruf yang dilengkapi dengan gambar - gambar, yang ditulis untuk berbagai keperluan [16].

Website Menurut Kadir 2004, *website* pertama kali dibuat oleh Tim Bamers Lee pada tahun 1990. *Website* ini dibangun menggunakan bahasa HTML dan menggunakan protokol komunikasi

HTTP yang terdapat pada lapisan aplikasi dalam referensi OSI. *Website* diakses oleh aplikasi yang menggunakan aplikasi yang disebut *browser* Internet [17].

2.11 HTML

HTML merupakan singkatan dari *Hyper Text Markup Language* adalah bahasa pemrograman dasar dalam membuat halaman web, HTML meliputi *Head*, *Body* dan memiliki *TAG* dan *Attribute*, walaupun dikatakan sebagai bahasa pemrograman, tidak dapat juga dikatakan bahwa HTML adalah bahasa pemrograman. Karena tidak memiliki apa yang dibutuhkan bahasa pemrograman yaitu logika, HTML hanya memberikan output, sehingga HTML dianggap sebagai fondasi atau struktur Web dan bahasa pemrogramannya adalah PHP dan *Javascript* [18].

Menurut Sidik dan Pohan, dokumen HTML adalah file teks murni yang dapat dibuat dengan editor teks apa pun. Dokumen ini disebut halaman web. Dokumen HTML adalah dokumen yang muncul di *browser* web pengguna Internet. Dokumen ini biasanya berisi informasi atau antarmuka sistem di Internet [19].

Menurut Hidayatullah dan Kawistara pada 2015, HTML adalah bahasa standar yang digunakan untuk menampilkan halaman web. Yang bisa dilakukan dengan HTML yaitu mengatur tampilan dari halaman web dan isinya, membuat tabel dalam halaman web, mempublikasikan halaman web secara online, dan membuat form yang bisa digunakan untuk menangani registrasi dan transaksi via web [20].

2.12 PHP

PHP dibuat oleh Rasmus Lerdorf, seorang *software* engineer dari Greenland sekitar tahun 1995. Pada awalnya, Rasmus hanya menggunakan PHP untuk mencatat pengunjung ke situs pribadinya. Inilah sebabnya mengapa bahasa ini disebut Personal Homepage Engine (PHP). Namun karena perkembangannya yang diapresiasi oleh masyarakat, ia merilis bahasa PHP tersebut ke publik [21].

PHP menurut Nazrul pada 2009, adalah singkatan dari "*Hypertext Processor*", yang merupakan bahasa scripting yang diinstal dalam HTML. Sebagian besar sintaks mirip dengan C,

Java, dan *Perl*, ditambah beberapa fungsi PHP tertentu. Tujuan utama menggunakan bahasa ini adalah untuk memungkinkan desainer web menulis halaman web dinamis dengan cepat. Sedangkan menurut Andi pada 2016, PHP adalah bahasa *scripting* sisi server yang dirancang untuk pengembangan web. Disebut bahasa pemrograman *server-side* karena PHP diproses di server. Hal ini berbeda dengan bahasa pemrograman *client-side* seperti *Javascript* yang diproses pada *web browser (client)* [16].

PHP termasuk bahasa pemrograman yang gratis dan bersifat open source. PHP dirilis dalam lisensi *PHP license*. Beberapa komponen dasar PHP antara lain adalah sintaks dasar PHP, memiliki beberapa aturan sintaks ketika membuat program PHP yaitu *PHP opening* dan *closing tag*, PHP memiliki komentar seperti pada bahasa pemrograman C, C++, dan lainnya. Terdapat konstanta dalam PHP didefinisikan menggunakan fungsi *define()* yang termasuk kedalam *function PHP*.

PHP memiliki beberapa fitur yang dapat digunakan oleh pengembang perangkat lunak. Berikut adalah beberapa fitur yang ada pada bahasa pemrograman PHP [22].

1. *Multi Paradigm Design*

Pada dasarnya PHP menggunakan paradigma pemrograman struktural, namun PHP juga dapat mendukung paradigma OOP (*Object Oriented Programming*).

2. *Open Source*

PHP adalah bahasa pemrograman *open source*. *Open source* sendiri adalah perangkat lunak atau sejenisnya yang dapat digunakan, dimodifikasi, dan didistribusikan secara bebas oleh siapa saja (baik telah dimodifikasi atau belum). Oleh karena itu, pengembang bebas mengubah bahasa pemrograman sesuai kebutuhan.

3. *Simplicity*

Sintaks PHP sederhana dan mudah dipelajari. Sintaks PHP mirip dengan bahasa pemrograman lain, sehingga pengembang dapat mempelajari PHP dengan cepat tanpa harus mempelajari sintaks dari awal.

4. *Framework Support*

Ada banyak *framework* yang tersedia dalam bahasa pemrograman PHP seperti CI (*Code Igniter*), *Laravel*, dan *Symfony*. Beragamnya *framework* yang ada memudahkan para *developer* untuk mengembangkan perangkat lunak karena mereka menggunakan *framework* tersebut. Mempercepat proses pengembangan perangkat lunak.

5. *Library Support*

Ada berbagai jenis *library* yang tersedia untuk pengembang PHP, termasuk ADOdb digunakan oleh bahasa pemrograman PHP dan banyak perpustakaan lain untuk mendukung *database* abstrak. *Library* dapat digunakan untuk mempercepat dan mengoptimalkan perangkat lunak yang dikembangkan.

6. *Extendable*

Pengembang perangkat lunak dapat menambahkan modul bahasa pemrograman lainnya. Contohnya adalah *Javascript*. *Javascript* sendiri adalah bahasa pemrograman turunan *Java* yang didedikasikan untuk pengembangan perangkat lunak berbasis skrip. *Javascript* dapat digunakan untuk membantu pengembang ketika modul yang diperlukan tidak tersedia, dan pengembang dapat menggunakan modul yang ada pada *javascript*.

7. *Portability*

PHP bersifat portabel, berjalan pada berbagai platform perangkat keras dan sistem operasi dan memiliki antarmuka yang sama di semua platform.

8. *Familiarity*

Bahasa pemrograman PHP digunakan oleh banyak situs web di seluruh Internet. Banyak pengembang perangkat lunak menggunakan bahasa pemrograman ini. Hal ini memudahkan pengembang untuk belajar dan berkembang.

2.13 Javascript

Javascript diperkenalkan pertama kali oleh Netscape pada tahun 1995. Pada awalnya bahasa ini dinamakan “*LiveScript*” yang berfungsi sebagai bahasa sederhana untuk *browser Netscape Navigator 2*. Pada masa itu bahasa ini banyak di kritik karena kurang aman, pengembangannya

yang terkesan terburu-buru dan tidak ada pesan kesalahan yang di tampilkan setiap kali membuat kesalahan pada saat menyusun suatu program. Kemudian sejalan dengan sedang giatnya kerjasama antara *Netscape* dan *Sun* (pengembang bahasa pemrograman *Java*) pada masa itu, maka *Netscape* memberikan nama “*Javascript*” kepada bahasa tersebut pada tanggal 4 desember 1995. Pada saat yang bersamaan *Microsoft* sendiri mencoba untuk mengadaptasikan teknologi ini yang mereka sebut sebagai “*Jscript*” di *browser Internet Explorer 3*. *Javascript* adalah bahasa yang berbentuk kumpulan skrip yang pada fungsinya berjalan pada suatu dokumen HTML, sepanjang sejarah internet bahasa ini adalah bahasa skrip pertama untuk web. Bahasa ini adalah bahasa pemrograman untuk memberikan kemampuan tambahan terhadap bahasa HTML dengan mengijinkan pengeksekusian perintah perintah di sisi *user*, yang artinya di sisi *browser* bukan di sisi server web. *Javascript* bergantung kepada *browser(navigator)* yang memanggil halaman web yang berisi skrip skrip dari *Javascript* dan tentu saja terselip di dalam dokumen HTML. *Javascript* juga tidak memerlukan kompilator atau penerjemah khusus untuk menjalankannya (pada kenyataannya kompilator *Javascript* sendiri sudah termasuk di dalam *browser* tersebut). Lain halnya dengan bahasa “*Java*” (dengan mana *Javascript* selalu di banding bandingkan) yang memerlukan kompilator khusus untuk menterjemahkannya di sisi *user/klien* [23].

Javascript merupakan suatu bahasa yang perkembangannya lambat di bandingkan dengan *Java* yang berkembang sangat cepat. Di *Javascript* kita tidak mungkin menyembunyikan kode skrip yang kita tulis, kode langsung di tulis di dalam dokumen HTML dan sangat mudah terlihat, sedangkan di *Java*, kode sudah berbentuk setengah terkompilasi (dalam bentuk applet) dan tidak mungkin terlihat dari dalam dokumen HTML, satu mesin virtual di sisi *user* yang bertanggung jawab untuk menterjemahkan program di dalam applet tersebut setiap kali halaman HTML yang memuat applet tersebut dipanggil oleh *browser*. Dibandingkan dengan applet *java* yang cukup lambat dibuka oleh *browser*, bisa kita katakan bahwa *Javascript* cukup cepat dipanggil (load) oleh navigator. *Javascript* sendiri merupakan bahasa yang mudah dipahami, dalam artian diperlukan skill novice atau dasar untuk mengerti bahasa ini, jika anda sudah terbiasa dan mengenal konsep bahasa pemrograman visual, maupun *Java* ataupun C, akan sangat mudah untuk memahami konsep *Javascript* [23].

Javascript adalah bahasa yang “*case sensitive*” artinya membedakan penamaan variabel dan fungsi yang menggunakan huruf besar dan huruf kecil, contoh variabel atau fungsi dengan nama

TEST berbeda dengan variabel dengan nama test. Dan yang terakhir seperti bahasa *Java* ataupun *C*, setiap instruksi diakhiri dengan karakter titik koma (;) [23].

Skrip dari *Javascript* terletak di dalam dokumen HTML. Kode tersebut tidak akan terlihat dari dalam jendela navigator anda, karena diantara tag tertentu yang memerintahkan navigator untuk memperlakukan bahwa skrip tersebut adalah skrip dari *Javascript* [23]. Contoh dari skrip yang menunjukkan bahwa skrip tersebut adalah skrip dari *Javascript* adalah sebagai berikut:

```
<script type="text/javascript">
Isi dari script
</script>
```

2.14 JSON

JSON (*Javascript* Object Notation) adalah format pertukaran data yang ringan, mudah untuk dibaca dan ditulis serta mudah diterjemahkan oleh mesin. JSON adalah format teks yang benar - benar independen namun menggunakan konvensi yang familiar bagi pemrogram yang termasuk dalam keluarga Bahasa *C*, termasuk *C*, *C++*, *C#*, *Java*, *Javascript*, *Perl* dan *Python*. Oleh karena sifat tersebut menjadikan JSON sebagai bahasa pertukaran data yang ideal [24].

JSON terbuat dari dua struktur utama, yaitu Kumpulan pasangan nama/nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (*object*), rekaman (*record*), struktur (*structure*), kamus (*dictionary*), tabel hash (*hash table*), daftar berkunci (**keyed list**), atau *associative array*. Daftar nilai yang diurutkan (an ordered list of values). Dalam kebanyakan bahasa, ini dinyatakan sebagai larik (*array*), vektor (*vector*), daftar (*list*), atau urutan (*sequence*) [24].

Struktur data ini disebut struktur data universal. Pada dasarnya, semua bahasa pemrograman modern mendukung struktur data ini dalam bentuk yang sama atau berbeda. Layak disebut demikian karena format datanya mudah dipertukarkan dengan bahasa pemrograman juga berdasarkan struktur data tersebut [24].

2.15 Tensorflow.js

Tensorflow.js adalah sebuah library yang dibangun diatas *deeplearn.js* untuk membuat modul deep learning langsung dari web *browser*. *Deep learning* sendiri adalah sebuah cabang dari

machine learning dan juga *artificial intelligence*. Dengan *Tensorflow.js* dapat membuat implementasi *Convolutional Neural Network (CNN)*, *Recurrent Neural Network (RNN)* dan lain sebagainya [25].

Tensorflow sendiri merupakan *library* yang ditulis dengan bahasa *C++* dan biasanya digunakan dengan bahasa pemrograman *Python*. Dengan adanya *tensorflow.js*, sekarang sudah bisa menggunakan beberapa fitur *tensorflow* di sisi web *browser* tanpa harus dibebani oleh instalasi yang cukup ‘rumit’ [25].

2.16 Database

Basis data (atau *database*) adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. *Database* digunakan untuk menyimpan informasi atau data yang terintegrasi dengan baik di dalam komputer [26].

Untuk mengelola *database* diperlukan suatu perangkat lunak yang disebut DBMS (*Database Management System*). DBMS merupakan suatu sistem perangkat lunak yang memungkinkan *user* (pengguna) untuk membuat, memelihara, mengontrol, dan mengakses *database* secara praktis dan efisien. Dengan DBMS, *user* akan lebih mudah mengontrol dan memanipulasi data yang ada [26].

Sedangkan RDBMS atau *Relationship Database Management System* merupakan salah satu jenis DBMS yang mendukung adanya *relationship* atau hubungan antar tabel. Di samping RDBMS, terdapat jenis DBMS lain, misalnya *Hierarchy DBMS*, *Object Oriented DBMS*, dsb. Beberapa *software* atau perangkat lunak DBMS yang sering digunakan dalam aplikasi program antara lain [26].

1. DB2
2. Microsoft SQL Server
3. Oracle
4. Sybase
5. Interbase
6. Terada
7. Firebird

8. MySQL

9. PostgreSQL

2.16.1 MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: *database management system*) atau DBMS yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis di bawah lisensi GNU *General Public License* (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL [26].

Tidak seperti PHP atau Apache yang merupakan *software* yang dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing, MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia yaitu MySQL AB. MySQL AB memegang penuh hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah: David Axmark, Allan Larsson, dan Michael "Monty" Widenius [26].

MySQL merupakan *software database* open source yang paling populer di dunia, dimana saat ini digunakan lebih dari 100 juta pengguna di seluruh dunia. Dengan kehandalan, kecepatan dan kemudahan penggunaannya, MySQL menjadi pilihan utama bagi banyak pengembang *software* dan aplikasi baik di platform web maupun desktop. Pengguna MySQL tidak hanya sebatas pengguna perseorangan maupun perusahaan kecil, namun perusahaan seperti *Yahoo!*, *Alcatel-Lucent*, *Google*, *Nokia*, *Youtube*, *Wordpress* dan *Facebook* juga merupakan pengguna MySQL. Berikut fitur-fitur MySQL antara lain [26].

1. *Relational Database System*. Seperti halnya *software database* lain yang ada di pasaran, MySQL termasuk RDBMS.
2. *Arsitektur Client-Server*. MySQL memiliki arsitektur *client-server* dimana server *database* MySQL terinstal di server. *Client* MySQL dapat berada di komputer yang sama dengan server, dan dapat juga di komputer lain yang berkomunikasi dengan server melalui jaringan bahkan internet.

3. Mengenal perintah SQL standar. SQL (*Structured Query Language*) merupakan suatu bahasa standar yang berlaku di hampir semua *software database*. MySQL mendukung SQL versi SQL:2003.
4. Mendukung Sub Select. Mulai versi 4.1 MySQL telah mendukung select dalam select (sub select).
5. Mendukung *Views*. MySQL mendukung *views* sejak versi 5.0
6. Mendukung *Stored Prosedured* (SP). MySQL mendukung SP sejak versi 5.0
7. Mendukung *Triggers*. MySQL mendukung *trigger* pada versi 5.0 namun masih terbatas. Pengembang MySQL berjanji akan meningkatkan kemampuan *trigger* pada versi 5.1.
8. Mendukung *replication*.
9. Mendukung transaksi.
10. Mendukung *foreign key*.
11. Tersedia fungsi GIS.
12. Free (bebas didownload)
13. Stabil dan tangguh
14. Fleksibel dengan berbagai pemrograman
15. Security yang baik
16. Dukungan dari banyak komunitas
17. Perkembangan *software* yang cukup cepat.

2.17 UML

UML (*Unified Modeling Language*) adalah sebuah bahasa yang berdasarkan grafik/gambar untuk memvisualisasi, menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan *software* berbasis OO (*Object-Oriented*). UML sendiri juga memberikan standar penulisan sebuah sistem blue print, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema *database*, dan komponen-komponen yang diperlukan dalam sistem *software*. UML adalah salah satu tool atau model untuk merancang pengembangan *software* yang berbasis *object oriented* [27].

UML sebagai sebuah bahasa yang memberikan *vocabulary* dan tatanan penulisan kata-kata dalam 'MS Word' untuk kegunaan komunikasi. Sebuah bahasa model adalah sebuah bahasa yang

mempunyai *vocabulary* dan konsep tatanan / aturan penulisan serta secara fisik mempresentasikan dari sebuah sistem. UML adalah sebuah bahasa standar untuk pengembangan sebuah *software* yang dapat menyampaikan bagaimana membuat dan membentuk model-model, tetapi tidak menyampaikan apa dan kapan model yang seharusnya dibuat yang merupakan salah satu proses implementasi pengembangan *software*. UML tidak hanya merupakan sebuah bahasa pemrograman visual saja, namun juga dapat secara langsung dihubungkan ke berbagai bahasa pemrograman, seperti *Java*, *C++*, *Visual Basic*, atau bahkan dihubungkan secara langsung ke dalam sebuah *object-oriented database*. Begitu juga mengenai pendokumentasian dapat dilakukan seperti; *requirements*, arsitektur, *design*, *source code*, *project plan*, *tests*, dan *prototypes*. Terdapat tiga elemen utama dari UML seperti building block. Tiga macam yang terdapat pada *building block* adalah benda, hubungan, dan bagan [27].

2.17.1 Use case diagram

Use case diagram terdiri dari lima grafik yang sangat sederhana yang mewakili sistem, aktor, *Use case*, asosiasi, dan dependensi proyek. Tujuan dari diagram adalah untuk memberikan penjelasan tingkat tinggi tentang hubungan antara sistem dan dunia luar. Ini adalah diagram yang sangat datar (yaitu, hanya memberikan tampilan tingkat permukaan, atau kotak hitam, dari sistem). Tampilan yang diwakili oleh diagram *Use case* untuk aplikasi ATM, misalnya, akan sesuai dengan layar utama ATM dan opsi menu yang tersedia di tingkat itu. Sistem ATM menawarkan kepada pengguna serangkaian opsi seperti penarikan, penyetoran, pertanyaan tentang saldo, dan transfer dana. Setiap opsi dapat diwakili oleh *Use case* yang terpisah. Pelanggan (di luar sistem) dikaitkan dengan setiap Kasus Penggunaan (di dalam sistem) yang akan digunakannya [28].

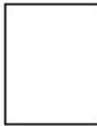
Enam elemen pemodelan membentuk diagram *Use case*: sistem, aktor, *Use case*, asosiasi, dependensi, dan generalisasi [28].

1. Sistem: Menetapkan batas sistem dalam kaitannya dengan aktor yang menggunakannya (di luar sistem) dan fitur yang harus disediakan (di dalam sistem).
2. Aktor: Peran yang dimainkan oleh seseorang, sistem, atau perangkat yang memiliki kepentingan dalam keberhasilan pengoperasian sistem.

3. *Use case*: Mengidentifikasi fitur kunci dari sistem. Tanpa fitur ini, sistem tidak akan memenuhi persyaratan pengguna/aktor. Setiap *Use case* mengungkapkan tujuan yang sistem harus dicapai.
4. Asosiasi: Mengidentifikasi interaksi antara aktor dan *Use case*. Setiap asosiasi menjadi sebuah dialog yang harus dijelaskan dalam sebuah narasi *Use case*. Setiap narasi dalam gilirannya menyediakan serangkaian skenario yang berfungsi sebagai kasus uji saat mengevaluasi analisis, desain, dan implementasi *Use case*.
5. *Dependency*: Mengidentifikasi hubungan komunikasi antara dua *Use case*.
6. *Generalization*: Mendefinisikan hubungan antara dua aktor atau dua *Use case* dimana satu *Use case* mewarisi dan menambah atau menimpa properti yang lain.

Tabel 2.1 Simbol Use case diagram

No	Gambar	Nama	Keterangan
1		Actor	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		Dependency	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (independent) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (independent).
3		Generalization	Hubungan dimana objek anak (descendent) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (ancestor).
4		Include	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
5		Extend	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.

6		Association	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		Sistem	Menspesifikasikan paket yang menampilkan sistm secara terbatas.
8		Use case	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
9		Collaboration	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		Note	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

2.17.2 Use case Scenario

Skenario *Use case* adalah salah satu jalur logis melalui *Use case*, satu kemungkinan urutan langkah dalam eksekusi *Use case*. Kasus Penggunaan dapat mencakup sejumlah skenario. Kumpulan skenario untuk satu *Use case* mengidentifikasi segala sesuatu yang bisa terjadi ketika *Use case* itu digunakan. Akibatnya, rangkaian skenario menjadi dasar untuk rencana pengujian Anda untuk Kasus Penggunaan. Saat desain aplikasi semakin dalam, rencana pengujian diperluas untuk menjaga pengujian tetap fokus pada ekspektasi awal untuk *Use case* yang dinyatakan dalam skenario.

Ketiga elemen ini adalah diagram *Use case*, narasi, dan skenario terdiri dari *Use case Model*. Sisa sesi ini dikhususkan untuk diagram *Use case*. Sesi 7 mencakup narasi *Use case*, dan Sesi 8 mencakup skenario *Use case* [28].

Tabel 2.2 Tabel Use case Scenario

<i>Use case Name</i>		
Goal Context		
Preconditions		
Succesful End Condition		
Failed End Condition		
Primary Actor		
Trigger		
Main Flow	Step	Action
Extensions	Step	Braching Action

2.17.3 Activity diagram

Pada dasarnya diagram *Activity* sering digunakan oleh flowchart. Diagram ini berhubungan dengan diagram Statechart. Diagram Statechart berfokus pada obyek yang dalam suatu proses (atau proses menjadi suatu obyek), diagram *Activity* berfokus pada aktifitas-aktifitas yang terjadi yang terkait dalam suatu proses tunggal. Jadi dengan kata lain, diagram ini menunjukkan bagaimana aktifitas-aktifitas tersebut bergantung satu sama lain [27].

Sebagai contoh, perhatikan proses yang terjadi. “Pengambilan uang dari bank melalui ATM.” Ada tiga aktifitas kelas (orang, dan lainnya) yang terkait yaitu: Customer, ATM, and Bank. Proses berawal dari lingkaran start hitam pada bagian atas dan berakhir di pusat lingkaran stop hitam/putih pada bagian bawah. Aktivitas digambarkan dalam bentuk kotak persegi.

Diagram *Activity* dapat dibagi menjadi beberapa jalur kelompok yang menunjukkan obyek yang mana yang bertanggung jawab untuk suatu aktifitas. Peralihan tunggal (*single transition*) timbul dari setiap adanya *activity* (aktifitas), yang saling menghubungi pada aktifitas berikutnya [27].

Sebuah *transition* (transisi) dapat membuat cabang ke dua atau lebih percabangan *exclusive transition* (transisi eksklusif). Label *Guard Expression* (ada di dalam []) yang menerangkan output (keluaran) dari percabangan. percabangan akan menghasilkan bentuk menyerupai bentuk intan. *transition* bisa bercabang menjadi beberapa aktifitas paralel yang disebut *Fork*. *Fork* beserta *join* (gabungan dari hasil *output fork*) dalam diagram berbentuk *solid bar* (batang penuh) [27].

Tabel 2.3 Simbol Activity diagram

No	Gambar	Nama	Keterangan
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi.
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Activity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan.
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran.

2.17.4 Class diagram

Diagram Class memberikan pandangan secara luas dari suatu sistem dengan menunjukkan kelas-kelasnya dan hubungan mereka. Diagram Class bersifat statis; menggambarkan hubungan apa yang terjadi bukan apa yang terjadi jika mereka berhubungan [27].

Diagram Class mempunyai 3 macam *relationships* (hubungan), sebagai berikut:

1. Association

Suatu hubungan antara bagian dari dua kelas. Terjadi association antara dua kelas jika salah satu bagian dari kelas mengetahui yang lainnya dalam melakukan suatu kegiatan. Di dalam diagram, sebuah association adalah penghubung yang menghubungkan dua kelas.

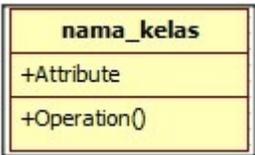
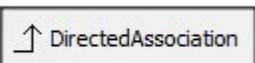
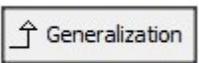
2. Aggregation

Suatu association dimana salah satu kelasnya merupakan bagian dari suatu kumpulan. Aggregation memiliki titik pusat yang mencakup keseluruhan bagian. Sebagai contoh: OrderDetail merupakan kumpulan dari Order.

3. Generalization

Suatu hubungan turunan dengan mengasumsikan satu kelas merupakan suatu superClass (kelas super) dari kelas yang lain. Generalization memiliki tingkatan yang berpusat pada superClass. Contoh: Payment adalah superClass dari Cash, Check, dan Credit.

Tabel 2.4 Simbol Class diagram

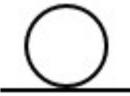
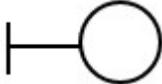
No	Gambar	Nama	Keterangan
1		Kelas	Kelas pada struktur sistem.
2		Antarmuka (<i>interface</i>)	Sama dengan konsep <i>interface</i> dalam pemrograman OOP.
3		Asosiasi (<i>Association</i>)	Relasi antar kelas dengan makna umum, asosiasi biasanya juga di sertai dengan <i>multiplicity</i> .
4		Asosiasi berarah (<i>Directed Association</i>)	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi berarah biasanya juga disertai dengan <i>multiplicity</i> .
5		Generalisasi (<i>Generalization</i>)	Relasi antar kelas dengan makna generalisasi-spesialisasi (Umum-Khusus)

6	 Dependency	Kebergantungan (<i>Dependency</i>)	Relasi antar kelas dengan makna kebergantungan antar kelas.
7	 Aggregation	Agregasi (<i>Aggregation</i>)	Relasi antar kelas dengan makna semua bagian (<i>Whole-part</i>).

2.17.5 Sequence diagram

Diagram Class dan diagram *Object* merupakan suatu gambaran model statis. Namun ada juga yang bersifat dinamis, seperti Diagram Interaction [27]. Diagram sequence merupakan salah satu diagram Interaction yang menjelaskan bagaimana suatu operasi itu dilakukan; message (pesan) apa yang dikirim dan kapan pelaksanaannya. Diagram ini diatur berdasarkan waktu. Obyek-obyek yang berkaitan dengan proses berjalannya operasi diurutkan dari kiri ke kanan berdasarkan waktu terjadinya dalam pesan yang terurut [27]. *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respon dari sebuah *event* untuk menghasilkan output tertentu. Diawali dari apa yang memicu aktivitas tersebut, proses, dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan.

Tabel 2.5 Simbol Sequence diagram

No	Gambar	Nama	Keterangan
1		Actor	Menggambarkan orang yang sedang berinteraksi dengan sistem.
2		Entity Class	Menggambarkan hubungan yang akan dilakukan.
3		Boundary Class	Menggambarkan sebuah gambar dari foem.
4		Control Class	Menggambarkan penghubung antara <i>boundary</i> dengan tabel.

5		A focus of Control & A life line	Menggambarkan tempat mulai dan berakhir message.
6		A message	Menggambarkan pengiriman pesan.