

BAB 2

TINJAUAN PUSTAKA

2.1 Landasan Teori

Landasan teori merupakan penjelasan berbagai konsep dasar dan teori-teori yang tersusun secara sistematis tentang hal – hal yang berkaitan dalam sebuah penelitian. Guna mendukung dalam penyusunan laporan, dibutuhkan landasan-landasan teori yang berkaitan dengan penelitian agar dalam penyusunannya searah dengan landasan teori tersebut. Berikut adalah landasan-landasan teori yang digunakan dalam penyusunan laporan ini.

2.1.1 Penyandang Disabilitas

Menurut Undang-Undang Nomor 19 Tahun 2011 Tentang Pengesahan Hak-Hak Penyandang Disabilitas, Penyandang disabilitas adalah orang yang memiliki keterbatasan fisik, mental, intelektual atau sensorik dalam jangka waktu lama yang dalam berinteraksi dengan lingkungan dan sikap masyarakatnya dapat menemui hambatan yang menyulitkan untuk berpartisipasi penuh dan efektif berdasarkan kesamaan hak. Terdapat beberapa istilah penyebutan menunjuk pada penyandang disabilitas, Kementerian Sosial menyebut dengan istilah penyandang cacat, Kementerian Pendidikan Nasional menyebut dengan istilah berkebutuhan khusus dan Kementerian Kesehatan menyebut dengan istilah Penderita cacat.

Menurut Reefani penyandang disabilitas dibagi menjadi beberapa jenis, yaitu Disabilitas Mental yang terdiri dari Mental Tinggi, Mental Rendah, Berkesulitan Belajar Spesifik. Disabilitas Fisik yang terdiri dari Tuna Daksa, Tuna Netra, Tunarungu, Tunawicara. Disabilitas Ganda seperti penyandang tuna netra dengan tunarungu sekaligus [6].

2.1.2 Tunawicara

Tunawicara merupakan individu yang mengalami kesulitan berbicara. Hal ini dapat disebabkan oleh kurang atau tidak berfungsinya alat-alat bicara, seperti rongga mulut, lidah, langit-langit dan pita suara. Selain itu, kurang atau tidak berfungsinya organ pendengaran, keterlambatan perkembangan bahasa, kerusakan

pada system saraf dan struktur *otot*, serta ketidakmampuan dalam control gerak juga dapat mengakibatkan keterbatasan dalam berbicara.

Tunawicara memiliki beberapa karakteristik antara lain memiliki suara sengau, cadel, bicara tidak jelas dan tidak mengeluarkan suara saat berbicara, cenderung pendiam, pandangan tertuju pada satu obyek, menggunakan komunikasi non verbal dan bahasa tubuh untuk mengungkapkan pendapat, pikiran dan keinginan, serta lebih memilih berkomunikasi secara tertulis [7].

2.1.3 Bahasa Isyarat

Bahasa isyarat merupakan bahasa yang mengutamakan komunikasi manual, yaitu menggunakan bahasa tubuh, tangan dan gerak bibir, bukan suara lisan. Kaum tunarungu adalah kelompok utama yang menggunakan bahasa ini, biasanya dengan mengkombinasikan bentuk tangan, orientasi dan gerak tangan, lengan, dan tubuh, serta ekspresi wajah untuk mengungkapkan pikiran mereka.

Bahasa isyarat unik dalam jenisnya di setiap negara. Bahasa isyarat bisa saja berbeda di negara-negara yang berbahasa sama. Contohnya, Amerika Serikat dan Inggris meskipun memiliki bahasa tertulis yang sama, memiliki bahasa isyarat yang sama sekali berbeda (American Sign Language dan British Sign Language).

Untuk di negara Indonesia sendiri terdapat dua jenis bahasa isyarat yaitu Sistem Isyarat Bahasa Indonesia (SIBI) dan Bahasa Isyarat Indonesia (BISINDO) namun sistem yang sekarang umum digunakan adalah Sistem Isyarat Bahasa Indonesia (SIBI) dimana sistem ini sama dengan bahasa isyarat yang diterapkan di Amerika (ASL - American Sign Language) [1].

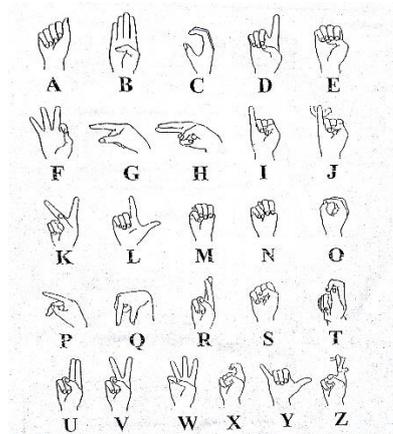
2.1.4 Sistem Isyarat Bahasa Indonesia (SIBI)

Sistem Isyarat Bahasa Indonesia (SIBI) merupakan bahasa isyarat yang diciptakan oleh Alm. Anton Widyatmoko mantan kepala sekolah SLB/B Widya Bakti Semarang bekerjasama dengan mantan kepala sekolah SLB/B di Jakarta dan Surabaya tanpa melalui musyawarah dan persetujuan dari Gerakan Kesejahteraan Tunarungu Indonesia atau GERKATIN yang pada akhirnya mengeluarkan sebuah produk kamus bernama SIBI. SIBI diterbitkan oleh pemerintah dan disebarluaskan melalui sekolah khususnya SLB/B sejak tahun 2001 [8].

Adapun *tolak ukur* dalam penentuan isyarat yang dibakukan dikemukakan dalam Kamus SIBI (Depdiknas, 2002 : xiv – xv) adalah sebagai berikut:

1. Sistem isyarat harus akurat dan konsisten mewakili sintaksis bahasa Indonesia.
2. Sistem isyarat harus mewakili satu kata dasar atau imbuhan.
3. Sistem isyarat harus mencerminkan situasi sosial, budaya, dan ekologi bahasa Indonesia serta dapat menghindari dari isyarat yang berkonotasi kurang etis.
1. Sistem isyarat harus mempertimbangkan perkembangan kemampuan dan kejiwaan siswa.
2. Sistem isyarat harus memperhatikan isyarat yang sudah banyak digunakan oleh anak tuna rungu.
3. Sistem isyarat harus mudah dipelajari dan digunakan.
4. Isyarat yang dirancang harus memiliki kelayakan dalam wujud dan maknanya.
5. Isyarat yang dirancang harus dapat dipakai pada jarak sedekat mungkin dengan mulut pengisyarat dan dengan kecepatan yang mendekati tempo berbicara yang wajar

Sistem Isyarat Bahasa Indonesia yang akan digunakan dalam pembangunan aplikasi ini adalah hanya huruf jari, diantaranya huruf a sampai z.



Gambar 2.1. SIBI Huruf

2.1.5 Kamera

Kamera adalah suatu set alat yang terintegrasi, digunakan sebagai alat untuk menghasilkan tampilan gambar dan bisa juga untuk merekam sebuah video. Kamera juga menjadi fitur pelengkap wajib pada ponsel atau *smartphone*. Kebanyakan kamera ponsel telah dilengkapi berbagai macam fitur, diantaranya [9]:

1. Fokus Tetap

Fitur dasar pada kamera ponsel yang diatur pada jarak tertentu oleh pengguna dan tidak dapat disesuaikan.

2. Fokus Otomatis

Fitur digital yang terdapat pada kamera ponsel yang memungkinkan untuk memperjelas foto atau gambar secara otomatis.

3. Makro

Makro adalah istilah yang digunakan untuk pengambilan gambar jarak dekat terhadap kamera ponsel.

4. Digital Zoom

Fitur ini digunakan untuk membuat subjek tampak lebih dekat terhadap kamera.

5. OIS (*Optical Image Stabilization*)

Merupakan mekanisme yang digunakan dalam kamera atau video kamera yang memiliki kemampuan untuk menstabilkan gambar yang di tampilkan atau direkam dengan memvariasikan jalur optik ke sensor.

6. Resolusi

Resolusi adalah suatu istilah kualitas yang mengacu pada jumlah piksel pada layar atau lensa kamera.

7. *Megapixel*

Mega merupakan satuan juta, *pixel* adalah suatu unit untuk mengukur resolusi kualitas gambar, dimana lebih banyak pixel berarti lebih jelas, lebih rinci, dan lebih detail. Sebuah pixel merupakan bagian terkecil dari informasi yang terdapat pada suatu gambar.

Terdapat fungsi lain dari kamera *smartphone*, diantaranya [10]:

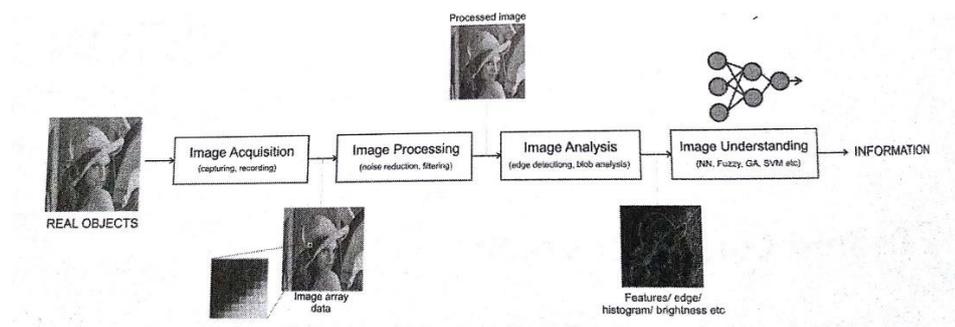
1. *Augmented Reality* (AR)

2. Mencari Tahu Informasi Suatu Benda
3. Alat Penerjemah
4. Scanner Dokumen
5. *QR Code Scanner*

2.1.6 *Computer Vision*

Computer Vision adalah suatu sistem otomatis analisa citra dan video oleh komputer dengan tujuan untuk memperoleh informasi dan pemahaman dari suatu objek. Secara umum, tujuan dari pengembangan *Computer Vision* ini adalah untuk menduplikasi kemampuan mata manusia baik dalam hal penangkapan/perekaman citra, analisa citra ataupun dalam pemahaman citra untuk diterapkan dalam sebuah mesin atau komputer. Dengan kata lain, *Computer Vision* adalah kemampuan sebuah mesin atau komputer untuk melihat dan memahami sebuah citra dengan sama atau bahkan melebihi dari kemampuan penglihatan manusia asli.

Dengan semakin berkembangnya teknologi *processor* komputer dan semakin murahnya sensor kamera CCD (*Coupled Cell Devices*), penelitian dan pengembangan *Computer Vision* ini berkembang cukup pesat akhir-akhir ini baik dilakukan oleh pemula, akademisi maupun kalangan profesional [3].



Gambar 2.2. Tahapan Dalam *Computer Vision*

Tahapan dalam *Computer Vision* dibagi menjadi 4, yaitu [3]:

1. Perekaman Citra (*Image Acquisition*)

Computer Vision menggunakan sensor visual untuk melakukan penangkapan data citra. Sensor visual yang digunakan pada umumnya adalah kamera. Hal ini sama dengan manusia yang merekam data visual sekitar dengan mata,

kemudian informasi visual tersebut diterjemahkan ke dalam suatu format yang kemudian dapat dimanipulasi oleh otak.

2. Pengolahan Data Citra (*Image Processing*)

Pengolahan data citra merupakan salah satu bagian dari proses dalam *Computer Vision*. Pengolahan ini khususnya membantu peningkatan dan perbaikan kualitas *image*, sehingga dapat dianalisa dan diolah lebih jauh secara lebih efisien. Selain itu, pengolahan data citra akan meningkatkan kualitas data citra yang dapat berupa peningkatan level sinyal-sinyal terhadap noise. Sinyal-sinyal tersebut adalah informasi yang akan merepresentasikan suatu objek dalam data citra. Sedangkan *noise* adalah segala bentuk interferensi, gangguan yang terdapat pada data citra.

3. Analisa dan Sintesa Data Citra (*Image Analysis*)

Dalam proses ini akan dilakukan eksplotasi beberapa karakteristik tertentu dari sebuah objek melalui data citra. Pada proses ini misalnya, pendeteksian tepi dan batasan objek dalam citra baik berdasarkan perbedaan warna, tingkat kecerahan, perbedaan kontras, bentuk, dan sebagainya. Pada pengembangan tingkat lanjut, pada tahapan ini biasanya dilakukan ekstrasi ciri sebagai identitas dari suatu citra.

4. Interpretasi dan Pemahaman Data Citra (*Image Understanding*)

Proses ini merupakan tahapan terakhir dalam *Computer Vision*. Pada umumnya, tahap *Computer Vision* menggunakan teknik kecerdasan buatan misalnya *Artificial Neural Network*, *Fuzzy System*, *Fuzzy Neural Network*, *Support Vector Machine*, *Principle Component Analysis*, *Evolutionary Algorithm*, dan yang paling mutakhir adalah dengan menggunakan *Deep Learning Network*. Selain itu, untuk interpretasi citra, juga bisa melakukan analisa secara *stokastik*, misalnya dengan menggunakan *Hidden Markov Model* atau *Gaussian Mixture Model*.

2.1.7 *Gesture Recognition*

Gesture dapat merujuk ke komunikasi non-verbal apa pun yang dimaksudkan untuk mengomunikasikan pesan tertentu. *Gesture* dapat didefinisikan sebagai gerakan fisik apa pun, besar atau kecil, yang dapat ditafsirkan oleh sensor gerak. Ini mungkin termasuk apa saja mulai dari jari telunjuk sampai tendangan lokomotif atau anggukan kepala ke cubitan atau lambaian tangan. Gerakan bisa lebar dan menyapu atau kecil dan berisi. Dalam beberapa kasus, definisi "*gesture*" juga dapat mencakup perintah suara atau lisan.

Gesture Recognition adalah jenis antarmuka pengguna komputasi persepsi yang memungkinkan komputer untuk menangkap dan menginterpretasi gerakan manusia sebagai perintah. Definisi umum *gesture recognition* adalah kemampuan komputer untuk memahami gerakan dan menjalankan perintah berdasarkan gerakan tersebut. Sebagian besar konsumen akrab dengan konsep melalui Wii Fit, X-box dan permainan PlayStation seperti "Just Dance" dan "Kinect Sports" [4].

2.1.8 **OpenCV**

OpenCV (*Open Source Computer Vision Library*) adalah sebuah library *open source* yang dikembangkan oleh intel yang fokus untuk menyederhanakan *programming* terkait citra digital. Di dalam OpenCV sudah mempunyai banyak fitur, antara lain : pengenalan wajah, pelacakan wajah, deteksi wajah, dan berbagai jenis metode AI (*Artificial Intelligence*). Dan menyediakan berbagai algoritma sederhana terkait *Computer Vision* untuk low level API. OpenCV merupakan *open source computer vision library* untuk bahasa pemrograman C/C++, dan telah dikembangkan ke python, java, matlab.

Intel meluncurkan versi pertama dari OpenCV pada 1999, dan awalnya memerlukan *library* dari *Intel Image Processing Library*. Kemudian depedensi tersebut akhirnya dihilangkan sehingga terciptalah OpenCV seperti yang sekarang sebagai *standalone library*. OpenCV mendukung *multiplatform*, dapat mendukung baik windows ataupun linux, dan sekarang telah mendukung MacOSX dan android.

Terdapat beberapa fitur yang dimiliki OpenCV, diantaranya adalah sebagai berikut [11]:

1. *Image and Video I/O*

Dapat membaca data gambar *file*, atau dari umpan video langsung dan dapat menciptakan *file* gambar maupun video.

2. *Computer Vision* secara umum dan pengolahan citra digital
Dapat melakukan eksperimen uji coba dengan berbagai standar algoritma *computer vision*. Termasuk deteksi garis, tepi, pucuk, proyeksi elips, *image pyramid* untuk pemrosesan gambar multi skala, pencocokan *template*, dan berbagai *transform* (*Fourier*, *Cosine Distrik*, *Distance Transform*) dan lain-lain.
3. Modul *Computer Vision High Level*
Terdapat kemampuan “*High Level*”, seperti kemampuan tambahan untuk deteksi wajah, pengenalan wajah, termasuk *optical flow*.
4. Metode untuk AI dan *Machine Learning*
Aplikasi *computer vision* terkadang memerlukan *machine learning* atau metode AI lainnya, beberapa metode tersebut tersedia dalam OpenCV *machine learning*.
5. Sampling gambar dan transformasi
Terdapat antar muka untuk substraksi subregion dari gambar, *random sampling*, *rotating*, dan lain-lain.
6. Metode untuk memperhitungkan pemodelan 3D
Metode ini bermanfaat untuk *mapping* dan *localization*, baik untuk *stereo camera*, ataupun satu kamera dengan berbagai sudut pandang.

2.1.9 Text

Text menurut kamus bahasa inggris yang berarti teks adalah wacana (berarti lisan) yang difiksasikan dalam bentuk tulisan. Dengan demikian jelas bahwa teks adalah fiksasi atau pelebagaan sebuah peristiwa wacana lisan dalam bentuk tulisan. Menurut Eriyanto dalam bukunya, Analisis Wacana, menyebutkan bahwa teks hampir sama dengan wacana, bedanya kalau teks hanya bisa disampaikan dalam bentuk tulisan saja, sedangkan wacana bisa disampaikan dalam bentuk lisan maupun tertulis [12].

Berikut adalah jenis-jenis atau macam-macam teks di dalam bahasa Indonesia [13]:

1. Teks Narasi

Narasi adalah jenis karangan fiksi atau tidak nyata yang menceritakan sebuah kisah dengan plot dan kronologi waktu yang disesuaikan dengan keinginan penulis. Tujuan dari penulisan narasi adalah sebagai media untuk menghibur pembaca melalui karangan fiksi penulis.

2. Teks Deskripsi

Merupakan jenis karangan yang berisi gambaran atau penjelasan terperinci dari sebuah objek. Penulisan karangan ini bertujuan untuk membuat pembaca seolah-olah merasakan, melihat, dan membayangkan apa yang dituliskan di dalam teks.

3. Teks Prosedur

Merupakan jenis karangan yang berisi langkah-langkah, tahapan, atau prosedur dari pembuatan atau pelaksanaan sebuah kegiatan. Jenis karangan prosedur yang sering ditemui misalnya resep masakan, tutorial pembuatan kerajinan tangan, dll. Tujuan penulisan teks prosedur adalah untuk berbagi cara atau petunjuk yang dimiliki oleh penulis agar pembaca lebih mudah memahami dan mengikuti melalui teks yang ditulis.

4. Teks Laporan

Merupakan jenis karangan yang berisi informasi atau laporan tentang sebuah topik. Biasanya jenis karangan ini banyak ditemukan di berita surat kabar. Karangan ini dapat ditulis dengan pengamatan umum atau sepintas, serta pengamatan dengan observasi langsung. Adapaun tujuan dari penulisan adalah untuk menyampaikan sebuah informasi kepada pembaca secara jelas dan umum berdasarkan hasil pengamatan.

5. Teks Eksposisi

Merupakan jenis karangan yang berisi paparan sebuah kejadian atau informasi yang bersifat padat dan akurat. Teks eksposisi biasanya ditemui dalam karangan non fiksi yang bersifat ilmiah.

2.1.10 *Speech*

Speech menurut kamus bahasa inggris yang berarti bahasa adalah kemampuan yang dimiliki manusia untuk berkomunikasi dengan manusia lainnya menggunakan tanda, misalnya kata dan gerakan. Atau alat untuk berinteraksi dan berkomunikasi, dalam arti alat untuk menyampaikan pikiran, gagasan, konsep atau perasaan. Pengertian bahasa secara umum adalah sistem lambang bunyi ujaran yang digunakan untuk berkomunikasi oleh masyarakat pemakainya. Bahasa yang baik berkembang berdasarkan suatu sistem, yaitu seperangkat aturan yang dipatuhi oleh pemakainya. Bahasa sendiri berfungsi sebagai sarana komunikasi serta sebagai sarana integrasi dan adaptasi [14].

Terdapat beberapa fungsi bahasa, diantaranya :

1. Untuk tujuan praktis, yaitu mengadakan hubungan dalam pergaulan sehari-hari.
2. Untuk tujuan artistik, yaitu manusia mengolah dan menggunakan bahasa dengan seindah-indahnya guna pemuasan rasa estetis manusia.
3. Alat untuk menyalurkan arti kepercayaan, norma dan nilai-nilai yang ada di masyarakat.
4. Sebagai alat untuk berpikir.
5. Sebagai alat kontrol sosial.
6. Untuk tujuan pembelajaran, sebagai media dalam mempelajari berbagai ilmu pengetahuan, baik itu yang masih berada pada ruang lingkup bahasa itu sendiri, ataupun diluar ruang lingkup bahasa, seperti pengetahuan sejarah dan ilmu pengetahuan yang lainnya.
7. Sebagai sarana memahami diri dan orang lain.
8. Sebagai sarana berekspresi.
9. Membangun karakter dan kecerdasan.
10. Untuk tujuan filologis, fungsi bahasa untuk mempelajari berbagai naskah tua untuk menyelidiki latar belakang dari sejarah manusia, perkembangan bahasa itu sendiri, dan lain sebagainya.
11. Sebagai pedoman untuk melihat adanya kenyataan di suatu masyarakat.

2.1.11 *Text To Speech*

Text to Speech atau sering disingkat TTS adalah sistem yang dapat mengubah suatu teks menjadi ucapan secara otomatis dengan cara fonetisasi, yaitu penyusunan fonem-fonem untuk membentuk ucapan. Dengan teknologi TTS, dimungkinkan sebuah komputer mampu berkomunikasi dan berinteraksi dengan manusia tidak hanya melalui tulisan, namun juga dalam bentuk lisan menggunakan bahasa yang digunakan sehari-hari. Secara umum pengertian TTS adalah sebuah sistem yang mengubah suatu teks menjadi bentuk ucapan. Menurut beberapa literatur, pengertian TTS adalah sebagai produksi ucapan secara otomatis melalui transkripsi *grapheme to phoneme* dari sebuah kalimat.

Pada prinsipnya, TTS terdiri atas dua sub sistem, yaitu *converter* teks ke fonem (*text to phoneme*) dan *converter fonem* ke ucapan (*phoneme to speech*). Bagian *converter* teks ke fonem berfungsi untuk mengubah kalimat masukan dalam suatu bahasa tertentu yang berbentuk teks menjadi rangkaian kode-kode bunyi yang biasanya direpresentasikan dengan kode fonem, durasi, serta *pitch*-nya [15].

2.1.12 *Application Programming Interface (API)*

Application Programming Interface (API) adalah sebuah teknologi untuk memfasilitasi pertukaran informasi atau data antara dua atau lebih aplikasi perangkat lunak. API adalah antarmuka virtual antara dua fungsi perangkat lunak yang saling bekerja sama, seperti antara sebuah *word processor* dan *spreadsheet*. Sebuah API mendefinisikan bagaimana cara programmer memanfaatkan suatu fitur tertentu dari sebuah komputer. API tersedia untuk sistem windowing, sistem file, sistem basis data dan sistem jaringan.

Perkembangan teknologi API berawal dari penciptaan sebuah subrutin sederhana yang menyediakan kemampuan *interoperability* dan *system modifiability* untuk mendukung pertukaran data antar beberapa aplikasi. Subrutin tersebut hanya mampu melaksanakan perhitungan matematis sederhana hingga terbentuk sebuah API pustaka perhitungan yang hampir selalu ada pada setiap bahasa pemrograman. Dari sebuah subrutin sederhana tersebut mulai bermunculan gagasan-gagasan bagaimana API harus dikembangkan khususnya sejalan dengan berkembangnya paradigma pemrograman berorientasi objek yang mengakibatkan sekumpulan

subrutin sejenis dapat dikumpulkan menjadi sebuah kelas pembungkus subrutin-subrutin tersebut [16].

2.1.13 Google Text To Speech API

Penggunaan *Google Text To Speech API* memungkinkan developer untuk mengkonversi teks ke dalam ucapan (suara). Layanan ini dapat diproses secara *online dan offline*. Untuk dapat menggunakan fitur tersebut, *developer android* dapat menggunakan *interface* dan *class* yang telah disediakan *Google API* (package `android.speech.tts`) [17].

Penggunaan secara online, pengguna diharuskan memiliki koneksi internet untuk dapat mengakses *database* pada server Google secara online. Bahasa yang didukung: Bengali (Bangladesh), Bengali (India), Kanton (Hong Kong), Ceko, Denmark, Belanda, Inggris (Australia), Inggris (India), Inggris (Inggris Raya), Inggris (Amerika Serikat), Estonia, Tagalog, Finlandia, Prancis, Jerman, Yunani, Hindi, Hungaria, Indonesia, Italia, Jepang, Khmer, Korea, China (China), China (Taiwan), Nepali, Norwegia, Polandia, Portugis (Brasil), Rumania, Rusia, Sinhala, Slovakia, Spanyol (Spanyol), Spanyol (Amerika Serikat), Swedia, Thai, Turki, Ukraina, dan Vietnam.

2.1.14 Perangkat Lunak

Perangkat lunak (*software*) adalah instruksi-instruksi (program computer) yang ketika dijalankan menyediakan fitur-fitur, fungsi-fungsi, dan kinerja-kinerja yang dikehendaki. Sebuah perangkat lunak juga sering disebut dengan sistem perangkat lunak. Sistem berarti kumpulan komponen yang saling terkait dan mempunyai satu tujuan yang ingin dicapai.

Sistem perangkat lunak berarti sekumpulan program yang ditulis untuk melayani program-program lain. (contoh : perangkat lunak untuk melakukan kompilasi (*compiler*) yang melakukan pemrosesan struktur-struktur informasi yang kompleks namun umumnya bersifat terbatas. User atau pemakai perangkat lunak adalah orang yang memiliki kepentingan untuk memakai atau menggunakan perangkat lunak untuk memudahkan pekerjaannya [5].

2.1.15 Pemrograman Berorientasi Objek

Berbeda dengan pemrograman terstruktur yang lebih fokus ke fungsi dan data secara terpisah, pada pemrograman berbasis objek (*object-oriented programming*) fungsi dan data bukanlah dua hal yang terpisah. Fungsi dan data menjadi satu kesatuan yang disebut objek aktif. OOP memandang sebuah program sebagai serangkaian objek yang bekerjasama untuk menyelesaikan suatu problem.

Pemrograman berorientasi objek bermaksud untuk memecahkan masalah pemrograman dengan pola pikir manusia. Pada pemrograman berbasis objek, kita membagi masalah pemrograman berdasarkan objek atau “sesuatu” benda. Objek ini diibaratkan sebagai makhluk hidup yang memiliki ciri-ciri fisik dan juga perilaku. Dalam pemrograman berorientasi objek, ciri-ciri fisik disebut atribut. Atribut ini berisi informasi tentang objek. Sementara perilaku disebut *metode*, yaitu bagaimana suatu objek bertindak atau melakukan sesuatu. Berikut karakteristik utama paradigma pemrograman berorientasi objek [18].

1. Menekankan pada data yang sedang beroperasi dan bukan pada fungsi atau prosedur.
2. Program dibagi ke dalam apa yang disebut dengan objek.
3. Data tersembunyi dan terlindungi dari fungsi atau prosedur yang ada di luar. Hal ini merupakan salah satu prinsip Java, yaitu teknik enkapsulasi.
4. Setiap objek dapat saling berkomunikasi.

Program desainnya dengan pendekatan *bottom-up*, yaitu membuat prosedur-prosedur untuk menyelesaikan tugas-tugas yang sederhana, kemudian menggabungkan prosedur-prosedur tersebut dalam prosedur yang lebih kompleks sampai fungsionalitas yang diinginkan tercapai.

2.1.16 Diagram UML

Salah satu model perancangan berorientasi objek yang saat ini paling sering digunakan di seluruh dunia adalah *Unified Modelling Language* (UML). UML sendiri adalah bahasa pemodelan untuk sistem atau perangkat lunak yang memiliki paradigma berorientasi objek.

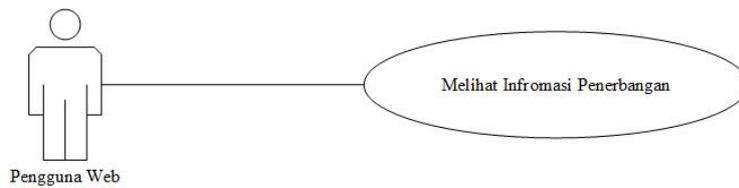
Unified Modelling Language (UML) adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. Dengan menggunakan UML, perancang atau pemodel dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. UML lebih cocok diterapkan pada piranti berorientasi objek seperti C++, Java, C#, dan sebagainya. Tetapi UML juga tetap dapat digunakan untuk modeling aplikasi prosedural semisal VB atau C.

UML versi 2.0 mencakup 13 macam diagram dan perangkat yang berfungsi untuk menggambarkan sistem informasi berorientasi objek dengan sangat lengkap dan rinci. Meski demikian, tidak selalu ke-13 diagram dan perangkat tersebut digunakan saat para pengembang berupaya mengemabangkan perangkat lunak berorientasi objek. Beberapa diantara digaram UML yang digunakan adalah *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, *Class Diagram*, dan *Collaboration Diagram* [19].

2.1.16.1 Use case diagram

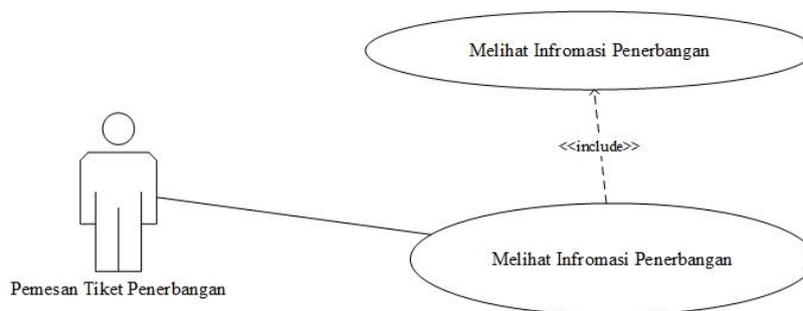
Use case diagram pada dasarnya digunakan untuk mendeskripsikan bagaimana entitas eksternal akan menggunakan sistem atau perangkat lunak. Entitas eskternal itu dapat berupa manusia atau sistem yang lain. Dalam diagram *use case*, entitas eksternal ini sering dinamakan sebagai *actor*. Deskripsi diagram *use case* ini lebih menekankan pada sistem dari sudut pandang penggunaannya dan juga menekankan pada interaksi yang terjadi di antara pengguna dengan sistem. *Use case* sangat membantu pengembang sistem untuk lebih jauh mendefinisikan ruang lingkup sistem serta batasan-batasannya.

Actor pada dasarnya adalah segala sesuatu yang berada di luar sistem atau perangkat lunak yang sedang dikembangkan. Setiap interaksi yang terjadi di antara *actordan* sistem dimodelkan sebagai *use case*. Contoh diagram *use case* sederhana dapat dilihat pada Gambar 2.3 [20].



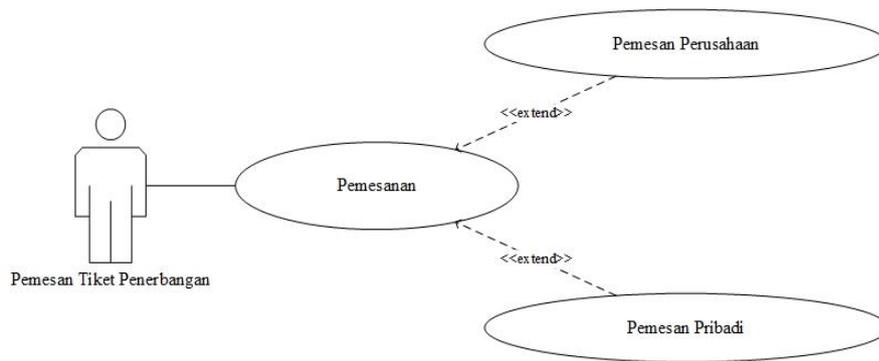
Gambar 2.3. Use Case Diagram Sederhana

Dalam diagram *use case*, sering ditemukan asosiasi yang bertipe *<include>*. Asosiasi *<include>* ini menunjukkan bahwa *use case* tertentu, misalnya “Melihat Informasi Penerbangan” harus dilakukan terlebih dahulu sebelum *use case* lainnya seperti “Memesan Kursi Pesawat” dilakukan. Terdapat asosiasi yang terjadi di antara *use case* “Melihat Informasi Penerbangan” dan *use case* “Memesan Kursi Pesawat”, sedangkan pemesan tiket penerbangan tidak dapat menggunakan *use case* “Memesan Kursi Pesawat” secara mandiri dari *use case* “Melihat Informasi Penerbangan”. Implementasi asosiasi *<include>* dalam kasus tersebut pada diagram *use case* dapat dilihat Gambar 2.4 [20].



Gambar 2.4. Use Case Diagram dengan Asosiasi Include

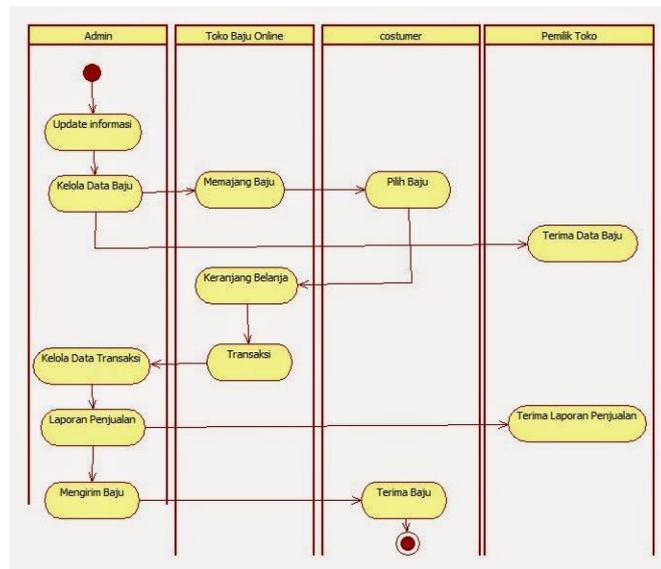
Selain asosiasi *<include>* dalam penggambaran juga dikenal asosiasi yang bertipe *<extend>*. Asosiasi *<extend>* ini hampir mirip dengan asosiasi bertipe *<include>*, hanya saja *use case* yang diasosiasikan sebagai *<extend>* tidak wajib dilaksanakan. Misalnya *actor* “Pemesan Tiket Penerbangan” dapat melakukan *use case* “Pemesanan” dengan terlebih dahulu melaksanakan *use case* “Pemesanan Perusahaan” atau “Pemesanan Pribadi”, tetapi tidak keduanya. Artinya, salah satu dari kedua *use case* ini harus dilaksanakan. Implementasi asosiasi *<extend>* dalam kasus tersebut pada diagram *use case* dapat dilihat Gambar 2.5 [20].



Gambar 2.5. Use Case Diagram dengan Asosiasi Extend

2.1.16.2 Activity Diagram

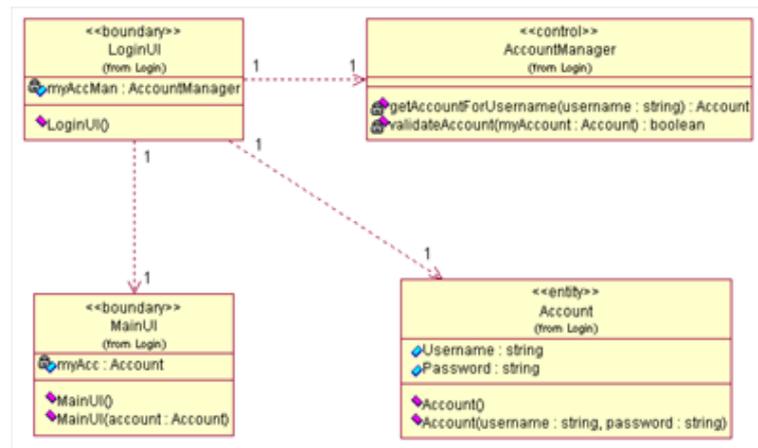
Sebuah *Activity Diagram* atau diagram aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana *actor* menggunakan sistem untuk melakukan aktivitas. Sama seperti *state*, standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. *Decision* digunakan untuk menggambarkan behaviour pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (*fork* dan *join*) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal. *Activity Diagram* dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu. Contoh dari *Activity Diagram* dapat dilihat pada Gambar 2.5 [21].



Gambar 2.6. Activity Diagram

2.1.16.3 Class Diagram

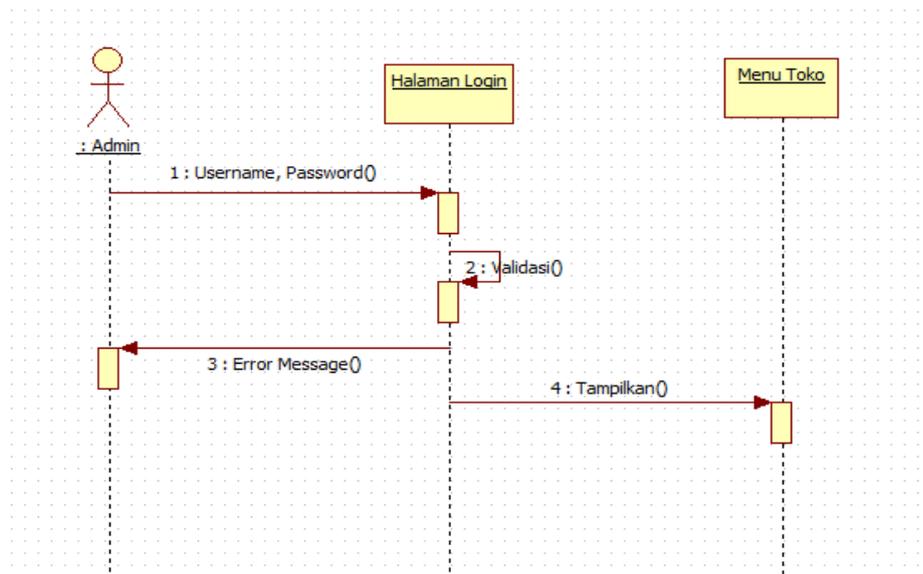
Class Diagram menggambarkan struktur statis dari kelas dalam sistem anda dan menggambarkan atribut, operasi dan hubungan antara kelas. *Class Diagram* membantu dalam memvisualisasikan struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. Selama tahap desain, *Class Diagram* berperan dalam menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat. *Class* memiliki tiga area pokok, yaitu nama (dan *stereotype*), atribut, dan metoda. Contoh *class diagram* dapat dilihat pada Gambar 2.6 [21].



Gambar 2.7. Class Diagram

2.1.16.4 Sequence Diagram

Sequence Diagram menjelaskan interaksi objek yang disusun berdasarkan urutan waktu. Secara mudahnya *Sequence Diagram* adalah gambaran tahap demi tahap, termasuk kronologi (urutan) perubahan secara logis yang seharusnya dilakukan untuk menghasilkan sesuatu sesuai dengan *use case diagram*. Contoh *Sequence Diagram* dapat dilihat pada Gambar 2.7 [21].



Gambar 2.8 Sequence Diagram

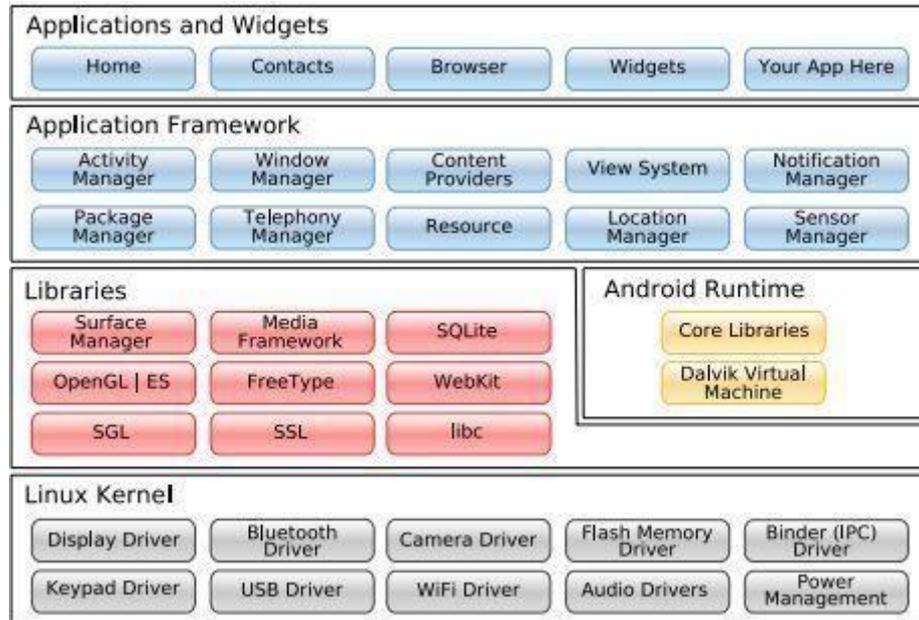
2.1.17 Android

Android merupakan suatu sistem operasi *mobile* yang berbasis pada sistem operasi Linux. Android pertama kali dikembangkan oleh perusahaan *startup* di California bernama Android, Inc., yang digawangi oleh Andy Rubin, Rich Miner, Nick Sears, dan Chris White. Pada 2005, Google membeli Android dan mengambil alih proses pengembangannya hingga saat ini. Google merilis versi beta Android SDK (*System Development Kit*) pada November 2007.

Android menawarkan pendekatan yang menyeluruh dalam pengembangan aplikasi. Artinya, satu aplikasi android yang dibangun dapat berjalan di berbagai perangkat yang menggunakan sistem operasi android baik itu *smartphone*, *smartwatch*, *tablet*, dan perangkat lainnya. Perkembangan teknologi android yang begitu pesat juga tidak dapat dilepas dari peranan AOSP (*Android Open Source Project*) yang bertanggung jawab dalam pengembangan sistem operasi Android dan dipimpin langsung oleh Google [22].

2.1.18 Arsitektur Android

Secara garis besar Arsitektur Android dapat dijelaskan dan dilihat pada gambar 2.8 dibawah ini. Pada gambar tersebut diperlihatkan bahwa terdapat empat tingkatan atau lapisan dalam arsitektur Android yang dimulai dari Linux Kernel sebagai lapisan terbawah, diikuti lapisan Libraries dan Android Runtime, lapisan Application Framework, hingga lapisan Applications dan Widgets [22].



Gambar 2.9. Arsitektur Android

Berikut ini adalah penjelasan dari gambar 2.8 mengenai arsitektur android :

1. *Applications dan Widgets*

Applications dan Widgets ini adalah *layer* dimana pengguna berhubungan dengan aplikasi saja, dimana biasanya pengguna men-*download* aplikasi kemudian melakukan instalasi dan menjalankan aplikasi tersebut. Di *layer* terdapat aplikasi inti termasuk klien email, program SMS, kalender, peta, browser, kontak, dan lain-lain. Semua aplikasi ditulis menggunakan bahasa pemrograman Java.

2. *Applications Frameworks*

Android adalah “*Open Development Platform*” yaitu Android menawarkan kepada pengembang atau memberi kemampuan kepada pengembang untuk membangun aplikasi yang bagus dan inovatif. Pengembang bebas untuk mengakses perangkat keras, akses informasi *resources*, menjalankan *service background*, mengatur *alarm*, menambahkan status *notificatios*, dan sebagainya. Pengembang memiliki akses penuh menuju *API framework* seperti yang dilakukan oleh aplikasi pada kategori inti. Arsitektur aplikasi dirancang supaya pengembang dengan mudah dapat menggunakan kembali komponen yang sudah digunakan (*reuse*). Sehingga pengembang bisa menyimpulkan *Applications Frameworks* ini adalah *layer* dimana para pembuat aplikasi melakukan pengembangan/pembuatan aplikasi

yang akan dijalankan di sistem operasi Android, karena pada *layer* inilah aplikasi dapat dirancang, seperti *content-providers* yang berupa sms dan lain sebagainya. Komponen-komponen yang termasuk didalam *Applications Frameworks* diantaranya *Views*, *Content Provider*, *Resource Manager*, *Notification Manager*, dan *Activity Manager*.

3. *Libraries*

Libraries ini adalah *layer* dimana fitur-fitur Android berada, biasanya para pembuat aplikasi kebanyakan mengakses *libraries* untuk menjalankan aplikasinya. Berjalan di atas kernel, *layer* ini meliputi berbagai *library C/C++* inti seperti sebagai *libc* dan *SSL*, serta:

1. *libraries* media untuk pemutaran media audio dan video
2. *libraries* untuk manajemen tampilan
3. *libraries Graphics* mencakup *SGL* dan *OpenGL* untuk grafis 2D dan
4. 3D
5. *libraries SQLite* untuk dukungan *database*
6. *libraries SSL* dan *WebKit* terintegrasi dengan *web browser* dan *security*
7. *libraries LiveWebcore* mencakup *modern web browser* dengan *engine embeded web view*
8. *libraries 3D* yang mencakup implementasi *OpenGL ES 1.0 API's*.

4. *Android Run Time*

Layer yang membuat aplikasi Android dapat dijalankan dimana dalam prosesnya menggunakan Implementasi Linux. *Dalvik Virtual Machine (DVM)* merupakan mesin yang membentuk dasar kerangka aplikasi Android. Didalam *Android Run Time* dibagi menjadi dua bagian yaitu:

1. *Core Libraries*: Aplikasi Android dibangun dalam bahasa java, sementara Dalvik sebagai virtual mesinnya bukan *Virtual Machine Java*, sehingga diperlukan sebuah *libraries* yang berfungsi untuk menerjemahkan bahasa java/c yang dihandle oleh *Core Libraries*.
2. *Dalvik Virtual Machine*: Virtual mesin berbasis register yang dioptimalkan untuk menjalankan fungsi-fungsi secara efisien, dimana merupakan

pengembangan yang mampu membuat linux kernel untuk *threading* dan manajemen tingkat rendah.

5. Linux Kernel

Linux *kernel* adalah *layer* dimana inti dari *operating sistem* dari Android itu berbeda. Berisi *file-file system* yang mengatur sistem *processing*, *memory*, *resource*, *drivers*, dan sistem-sistem *operating sistem android* lainnya. Linux *karnel* yang digunakan android adalah linux *karnel 2.6*.

2.1.19 Android Software Development Kit (SDK)

Android SDK untuk membantu proses pembangunan aplikasi Android, Android SDK berisikan beragam *Libraries* dan *Build tools*. Android merupakan subset perangkat lunak untuk ponsel yang meliputi sistem operasi, middleware, dan aplikasi kunci yang di-release oleh Google [22]. Beberapa fitur Android yang paling penting diantaranya:

- 1) Framework Aplikasi yang mendukung penggantian komponen dan reusable.
- 2) Mesin Virtual Dalvik dioptimalkan untuk perangkat mobile.
- 3) Integrated browser berdasarkan engine open source WebKit.
- 4) Grafis yang dioptimalkan dan didukung oleh libraries grafis 2D, grafis 3D berdasarkan spesifikasi OpenGL ES 1.0 (Opsional akselerasi hardware).
- 5) SQLite untuk penyimpanan data.
- 6) Media Support yang mendukung audio, video, dan gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF), GSM Telephony (tergantung hardware).
- 7) Bluetooth, EDGE, 3G, dan WiFi (tergantung hardware).
- 8) Kamera, GPS, kompas, dan accelerometer (tergantung hardware).

Lingkungan Development yang lengkap dan kaya termasuk perangkat emulator, tools untuk debugging, profil, dan kinerja memori.

2.1.20 Android Studio

Android Studio merupakan salah satu *modern code editor* yang sangat membantu dalam pembangunan suatu aplikasi Android. Sebagai IDE (*Integrated Development Environment*) resmi yang disarankan oleh Google dalam pembangunan aplikasi Android, Android Studio menawarkan banyak fitur dan *tools* yang dapat digunakan oleh pengembang aplikasi Android.

Android Studio adalah IDE pemrograman Android resmi dari Google yang dikembangkan dari IntelliJ. Dikembangkan di atas IntelliJ IDEA besutan JetBrains, Android Studio dirancang khusus untuk pengembangan Android. IDE Ini tersedia untuk digunakan pada sistem operasi Windows, Mac OS X dan Linux. Android Studio diumumkan pada 16 Mei 2013 pada konferensi Google I/O oleh Product Manager Google, Ellie Powers. [23]

Setiap proyek di Android Studio berisi satu atau beberapa modul dengan file kode sumber dan file sumber daya. Jenis-jenis modul mencakup: Modul aplikasi Android, Modul Pustaka dan Modul Google App Engine. Android studio menawarkan banyak fitur yang memudahkan dalam pembangunan aplikasi android seperti [24] :

1. Sistem versi berbasis Gradle yang fleksibel
2. Emulator yang cepat dan kaya fitur
3. Lingkungan yang menyatu untuk pengembangan bagi semua perangkat Android
4. Instant Run untuk mendorong perubahan ke aplikasi yang berjalan tanpa membuat APK baru
5. Template kode dan integrasi GitHub untuk membuat fitur aplikasi yang sama dan mengimpor kode *contoh*
6. Alat pengujian dan kerangka kerja yang ekstensif
7. Alat Lint untuk meningkatkan kinerja, kegunaan, kompatibilitas versi, dan masalah-masalah lain
8. Dukungan C++ dan NDK

Dukungan bawaan untuk Google Cloud Platform, mempermudah pengintegrasian Google Cloud Messaging dan App Engine.

