

## **BAB 2**

### **TINJAUAN PUSTAKA**

#### **2.1 SLBN Kabupaten Cirebon**

SLBN Kab. Cirebon adalah satu diantara SLB yang ada di kab. Cirebon yang berstatus negeri. Beralamat di Jl. AR Hakim No 33, Sindang Laut, Kec. Lemah Abang, Kab. Cirebon Prov. Jawa Barat, Dengan luas tanah 3960 M2. SLBN Kab. Cirebon memiliki 21 orang tenaga pengajar dan total siswa 144 orang mulai dari satuan pendidikan TKLB sampai dengan SMALB dengan jenis Kekhususan beragam seperti tunanetra, tunarungu, tunagharita, tunadaksa dan autis.

#### **2.2 Visi dan Misi Sekolah**

Visi adalah serangkaian kata yang menunjukkan impian, cita-cita atau nilai inti sebuah organisasi, perusahaan, atau instansi[2]. Visi merupakan tujuan masa depan sebuah instansi, organisasi, atau perusahaan. Visi juga adalah pikiran-pikiran yang ada di dalam benak para pendiri. Pikiran-pikiran tersebut adalah gambaran tentang masa depan yang ingin dicapai. Adapun visi dari SLBN Kab. Cirebon adalah sebagai berikut : Dengan iman dan taqwa SLBN Kan. Cirebon unggul dalam bidang akademik dan keterampilan kecakapan hidup pada tahun 2018.

Misi Adalah tahapan-tahapan yang harus dilalui untuk mencapai visi tersebut[2]. Selain itu, misi juga merupakan deskripsi atau tujuan mengapa perusahaan, organisasi, atau instansi tersebut berada di tengah-tengah masyarakat. isi juga bisa dikatakan sebagai Penjabaran sebuah visi. Jika visi hanya dituliskan dalam satu kalimat saja, maka misi akan dijabarkan dengan beberapa kalimat yang mudah untuk dipahami pembaca atau siapa saja yang melihatnya. Adapun Misi dari SLBN Kab. Cirebon adalah :

1. Optimalisasi pembelajaran pendidikan agama dan budi pekerti.
2. Menciptakan kondisi suasana yang harmonis, religius, dan berkarakter dilingkungan sekolah.

3. Mengembangkan potensi peserta didik dalam bidang akademik dan keterampilan untuk bekal di masyarakat.

Menumbuhkembangkan pengetahuan, percaya diri dan kemandirian peserta didik sehingga berguna bagi diri dan lingkungannya.

### **2.3 Strategi Sekolah**

Strategi adalah cara atau proses yang digunakan untuk tercapainya sebuah tujuan[2]. Strategi merupakan tindakan yang bersifat incremental (senantiasa meningkat) dan terus menerus, yang dilakukan berdasarkan sudut pandang tentang tujuan yang diharapkan. Berikut adalah Strategi dari SLBN Kab. Cirebon :

1. Meningkatkan kualitas pendidikan agama dan budi pekerti.
2. Meningkatkan pengetahuan bidang akademik yang berkarakter dan ketrampilan.
3. Meningkatkan kompetensi guru melalui pendidikan dan latihan.
4. Menjalin kerjasama warga sekolah dan stake holder yang terkait.
5. Menjalin kerjasama dengan Departemen Agama, Dinas Perdagangan dan Perindustrian.

### **2.4 Tujuan**

Menurut kamus besar bahasa indonesia, tujuan adalah arah atau yang dituju[2]. Dan arti tujuan dalam kelembagaan adalah tujuan atau kualifikasi yg diharapkan dimiliki murid setelah dia menerima atau menyelesaikan program pendidikan pd lembaga pendidikan tertentu. Berikut adalah tujuan dari SLBN kab. Cirebon :

1. Menumbuh kembangkan rasa cinta dan kasih sayang terhadap ciptaan tuhan.
2. Terbentuknya peserta didik yang berbudi dan berakhlak mulia.
3. Terwujudnya guru yang memiliki kompetensi dan dedikasi tinggi.
4. Meningkatkan Pengetahuan dan keterampilan sesuai potensi yang dimiliki.
5. Terwujudnya peserta didik yang mampu menerapkan keterampilan berkebun.

6. Terwujudnya peserta didik yang terampil berkebun sesuai bekal dalam kehidupan sehari-hari.

## **2.5 Penyandang Disabilitas**

Menurut Kamus Besar Bahasa Indonesia penyandang diartikan dengan orang yang menyandang (menderita) sesuatu. Sedangkan disabilitas merupakan kata bahasa Indonesia yang berasal dari kata serapan bahasa Inggris disability (jamak: disabilities) yang berarti cacat atau ketidakmampuan[2].

Menurut Undang-Undang Nomor 19 Tahun 2011 Tentang Pengesahan Hak-Hak Penyandang Disabilitas, penyandang disabilitas yaitu 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.

Berdasarkan pengertian di atas dapat disimpulkan bahwa orang berkebutuhan khusus atau penyandang disabilitas adalah orang yang hidup dengan keterbatasan atau ketidakmampuan baik fisik, mental atau sensorik yang dapat menghambat aktifitas sehari-hari. Berdasarkan hal tersebut maka penyandang disabilitas memerlukan pelayanan khusus untuk mendapatkan hak-haknya.

## **2.6 Tunanetra**

Menurut Daniel P. Hallahan, James M. Kauffman, dan Paige C. Pullen Tunanetra adalah seseorang yang memiliki ketajaman visual 20/200 atau kurang pada mata/penglihatan yang lebih baik setelah dilakukan koreksi (misalnya kacamata) atau memiliki bidang penglihatan begitu sempit dengan diameter terlebar memiliki jarak sudut pandang tidak lebih dari 20 derajat[3]. Seorang anak yang mengalami gangguan ataupun kerusakan penglihatan akan mengalami berbagai hambatan. Anak yang mengalami ketidakmampuan melihat adalah anak yang mempunyai gangguan atau kerusakan dalam penglihatannya akan mengalami hambatan dalam prestasi belajar secara optimal dan harus dilakukan penyesuaian dalam proses pembelajarannya.

Berdasarkan pendapat diatas dapat disimpulkan bahwa seorang Anak tunanetra adalah anak yang mengalami keterbatasan penglihatan secara keseluruhan (the blind) atau secara sebagian (low vision) yang akan menghambat mendapatkan informasi secara visual sehingga mengalami hambatan dalam pembelajaran dan prestasi belajar.

### **2.6.1 Karakteristik Umum Tunanetra**

Dalam buku Psikologi Pendidikan, karya Jeanne Ellis Ormrod, disebutkan bahwa siswa yang mengalami gangguan visual biasanya memiliki beberapa atau semua dari karakteristik-karakteristik berikut ini:

- 1 Indera lainnnya berfungsi normal (pendengaran, sentuhan, dan sebagainya).
- 2 Secara umum memiliki kemampuan belajar yang sama dengan siswa normal.
- 3 Perbendaharaan kata dan pengetahuan umu yang lebih terbatas, sebagian disebabkan oleh terbatasnya kesempatan untuk mengalami dunia luar melalui fasilitas pendidikan (misalnya: kurang mampu melihat peta, film, dan materimateri visual lainnya).
- 4 Menurunnya kapasitas untuk meniru perilaku orang lain.
- 5 Tidak mampu mengamati bahasa tubuh orang lain dan tanda-tanda nonverbal yang membuat mereka terkadang keliru memahami pesan-pesan orang lain.
- 6 Merasa bingung dan cemas (khususnya di tempat orang lalu lalang seperti di ruang makan atau taman bermain) karena memiliki pengetahuan yang terbatas mengenai peristiwa-peristiwa yang sedang berlangsung)[4].

### **2.6.2 Klasifikasi Tunanetra**

Ada beragam klasifikasi pada tuna netra, namun pada dasarnya tuna netra dibagi menjadi dua kelompok, yaitu kurang penglihatan (low vision) dan buta total (totally blind).

Kurang penglihatan (low vision), yakni mereka yang memiliki pandangan yang kabur ketika melihat suatu objek, sehingga untuk mengatasi permasalahan penglihatannya, penderita tunanetra jenis low vision perlu menggunakan kacamata

atau kotak lensa. Sedangkan, yang dimaksud buta total (totally blind), yakni mereka yang sama sekali tidak mampu melihat rangsangan cahaya dari luar. Menurut Direktorat Pembinaan Sekolah Luar Biasa, tuna netra secara garis besar dapat dikelompokkan berdasarkan empat kategori, yaitu 1) waktu terjadinya ketunanetraan, 2) kemampuan daya penglihatan, 3) pemeriksaan klinis, dan 4) kelainan/kekelainan pada mata. Keempat klasifikasi ketunanetraan itu dijelaskan sebagai berikut:

- 1 Berdasarkan waktu terjadinya ketunanetraan, ada lima kategori tuna netra, yaitu sebagai berikut:
  - 1 Tuna netra sebelum dan sejak lahir, yakni mereka yang sama sekali tidak memiliki pengalaman penglihatan.
  - 2 Tuna netra setelah lahir atau pada usia kecil, yaitu mereka yang telah memiliki kesan-kesan serta pengalaman visual, tetapi belum kuat dan mudah terlupakan.
  - 3 Tuna netra pada usia sekolah atau pada masa remaja, yaitu mereka yang telah memiliki kesan-kesan visual dan meninggalkan pengaruh yang mendalam terhadap proses perkembangan pribadi.
  - 4 Tuna netra pada usia dewasa, yaitu pada umumnya mereka yang dengan segala kesadaran mampu melakukan latihan-latihan penyesuaian diri.
  - 5 Tuna netra dalam usia lanjut, yaitu sebagian besar sudah sulit mengikuti latihan-latihan penyesuaian diri.
- 2 Berdasarkan kemampuan daya penglihatan, tuna netra ddibagi menjadi 3 kategori sebagai berikut:
  - 1 Tuna netra ringan (defective vision/low vision), yakni mereka yang memiliki hambatan dalam penglihatan, tetapi mereka yang mengikuti program-program pendidikan dan mampu melakukan pekerjaan/kegiatan yang menggunakan fungsi penglihatan.
  - 2 Tuna netra setengah berat (partially sighted), yakni mereka yang kehilangan sebagian daya penglihatan, hanya dengan menggunakan kaca pembesar mampu mengikuti pendidikan biasa atau mampu membaca tulisan yang bercetak tebal.

- 3 Tuna netra berat (totally blind), yakni mereka yang sama sekali tidak bisa melihat.
- 3 Berdasarkan pemeriksaan klinis, tuna netra dibagi menjadi 2 kategori sebagai berikut:
  - 1 Tuna netra yang memiliki ketajaman penglihatan kurang dari 20/200 dan atau memiliki bidang penglihatan kurang dari 20 derajat.
  - 2 Tuna netra yang masih memiliki ketajaman penglihatan antara 20/70 sampai dengan 20/200 yang dapat lebih baik melalui perbaikan.
- 4 Berdasarkan kelainan pada mata, tuna netra dibagi menjadi 3 kategori sebagai berikut:
  - 1 Myopia, yaitu penglihatan jarak dekat, bayangan tidak terfokus dan jatuh di belakang retina. Penglihatan akan menjadi jelas kalau objek didekatkan. Untuk membantu proses penglihatan pada penderita myopia digunakan kacamata proyeksi dengan lensa negative.
  - 2 Hyperopia, yaitu penglihatan jarak jauh, bayangan tidak terfokus, dan jatuh tepat di retina. Penglihatan akan menjadi jelas jika objek dijauhkan. Untuk membantu proses penglihatan pada penderita hyperopia digunakan kacamata koreksi dengan lensa positif.
  - 3 Astigmatisme, yaitu penyimpangan atau penglihatan kabur yang disebabkan karena ketidakberesan pada kornea mata atau pada permukaan lain pada bola mata sehingga bayangan benda baik pada jarak dekat maupun jauh tidak terfokus jatuh pada retina. Untuk membantu proses penglihatan pada penderita astigmatisme digunakan kacamata koreksi dengan lensa silindris[3].

### ***2.7 Optical Character Recognition(OCR)***

OCR (Optical Character Recognition) atau sering disebut dengan Pengenalan Pola karakter yang merupakan salah satu cabang dari pengenalan pola. Salah satu metode pengenalan pola angka atau huruf banyak digunakan algoritma Neural Network Backpropagation, yang sistem kerjanya seperti otak manusia yang dapat

mengenali pola setelah mendapatkan pelatihan atau pengecekan terhadap beberapa contoh terlebih dahulu.

Teknologi OCR menggunakan dua metode, yaitu *matric matching* dan *feature extraction*. Dari dua metode ini, *matric matching* adalah teknik yang paling simple dalam mendeteksi karakter. Selain itu juga terbatas, sehingga keakuratannya kerap lebih rendah dibandingkan *feature extraction*.

*Matric Matching* dan *Feature Extraction* Metode pertama dan yang paling simple adalah *Matrix Matching*. Lebih dikenal sebagai *pattern matching (String Matching)*. Ini dikarenakan scanner akan mendeteksi karakter yang dipindai lalu menyamakan dengan database dari pola atau outline dari setiap karakter huruf yang disimpannya dalam bentuk *bitmap image*. Jadi ketika scanner memindai sebuah huruf maka aplikasi akan mendeteksi bentuk huruf tersebut dan menyamakan bentuk, sudut, dan pola karakter dari gambar *bitmap* yang ada di database. Jika ada yang cocok maka huruf yang dipindai tersebut akan diidentifikasi sebagai huruf yang sama dalam database. Hanya saja teknik ini memiliki keterbatasan, karena database ini memiliki keterbatasan akan jumlah variasi bentuk karakter. Karena, font saat ini beraneka ragam, dan bentuknya ada yang semakin kompleks dan beraneka pola.

Metode kedua, *Feature Extraction* lebih pintar. Dikenal juga sebagai *ICR* atau *Intelligent Character Recognition* atau *topological feature analysis*. Ini adalah metode pengenalan karakter yang tidak tergantung dari template karakter yang dimiliki aplikasi. Aplikasi dengan metode ini akan mendeteksi identitas sebuah karakter dari ruang kosong, bentuk yang berdekatan, garis diagonal, perpotongan dan sebagainya. Sehingga pengidentifikasiannya lebih kompleks prosesnya. *Feature Extraction* biasanya digunakan untuk mendeteksi karakter yang bentuknya tak standar, seperti tulisan tangan atau huruf-huruf berornamen. *Matric Matching* cocoknya untuk document yang menggunakan jenis huruf yang umum, seperti *Times New Romans*, *Arial*, atau font yang memiliki bentuk dasar karakter huruf lainnya. Database karakter ini biasanya meliputi rangkaian dasar karakter angka dari 0-9 dan huruf a-z.

Cara kerja OCR terdiri dari modul yang terpisah yang digunakan dalam proses pengenalan, modul-modul tersebut adalah:

#### 1) Pra Pengolahan (Preprocessing)

Merupakan suatu proses untuk menghilangkan bagian-bagian yang tidak diperlukan pada gambar input untuk proses selanjutnya. Contoh preprocessing adalah noise filtering.

#### 2) Klasterisasi (Clusterization)

Modul klasterisasi bertugas untuk menemukan karakter individu dalam gambar dan menghitung urutan karakter yang benar dalam teks.

#### 3) Ekstraksi Fitur (Feature extraction)

Tujuan dari modul ini adalah untuk mengekstrak fitur unik dari karakter individu sehingga dapat diakui oleh modul klasifikasi. Jika kita membandingkan sistem OCR dengan manusia, kita dapat mengatakan bahwa modul ini memainkan peran mata manusia.

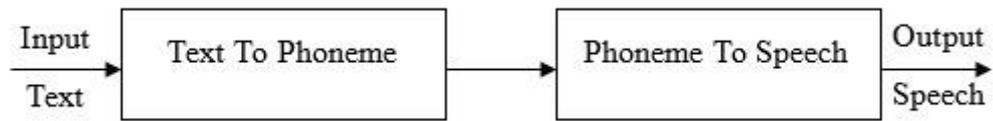
#### 4) Klasifikasi (Classification)

Pada tahap terakhir sistem OCR mencoba untuk mengenali karakter menggunakan informasi tentang fitur karakter yang telah diekstraksi pada tahap sebelumnya. Meskipun dapat dilakukan secara statistik dengan melibatkan perbandingan konsisten karakter dari database, hal ini dipilih untuk mencoba bagaimana jaringan saraf melakukan pengenalan pada tahap ini[5].

## ***2.8 Text To Speech***

Text to Speech (TTS) diartikan sebagai proses pengubahan teks menjadi audio digital dan diucapkan. Pengucapan ini dapat berupa pengiriman audio digital tersebut ke pengeras suara computer atau menyimpan hasil pengubahan tersebut untuk diputar nanti. Tugas sistem TTS secara umum dapat dibagi dalam 2 bagian besar, analisa teks dan sintesa ucapan. Analisa teks mentransformasi teks masukan menjadi representasi linguistik, selanjutnya bagian sintesis ucapan mentransformasi representasi linguistik tersebut menjadi gelombang sinyal ucapan[6].

Sistem *Text to Speech* terdiri dari dua sub sistem , yaitu: Bagian Konverter Teks ke Fonem (*Text to Phoneme*) dan bagian Konverter Fonem to Ucapan (*Phoneme to Speech*).



**Gambar 2.1** Skema *Text To Speech*

Berikut ini adalah penjelasan dari gambar 2.1 :

1) Bagian Konverter Teks ke Fonem (*Text to Phoneme*)

Bagian Konverter 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. Bagian ini bersifat sangat *language dependant*. Untuk suatu bahasa baru, bagian ini harus dikembangkan secara lengkap khusus untuk bahasa tersebut.

2) Bagian Konverter Fonem ke Ucapan (*Phoneme to Speech*)

Bagian Konverter Fonem ke Ucapan menerima masukan berupa kode-kode fonem serta *pitch* dan durasi yang dihasilkan oleh bagian sebelumnya. Berdasarkan kode-kode tersebut, bagian Konverter Fonem ke Ucapan akan menghasilkan bunyi atau sinyal ucapan yang sesuai dengan kalimat yang ingin diucapkan. Ada beberapa alternatif teknik yang dapat digunakan untuk implementasi bagian ini. Dua teknik yang banyak digunakan adalah *formant synthesizer* serta *diphone concatenation*. Sub sistem ini harus memiliki pustaka setiap unit ucapan dari suatu bahasa[6].

*Formant synthesizer* bekerja berdasarkan suatu model matematis yang akan melakukan komputasi untuk menghasilkan sinyal ucapan yang diinginkan. *Synthesizer* jenis ini telah lama digunakan pada berbagai aplikasi. Walaupun dapat menghasilkan ucapan dengan tingkat kemudahan interpretasi yang baik, *synthesizer* ini tidak dapat menghasilkan ucapan dengan tingkat kealamian yang tinggi.

*Synthesizer* yang menggunakan teknik *diphone concatenation* bekerja dengan cara menggabung-gabungkan segmen-segmen bunyi yang telah direkam sebelumnya. Setiap segmen berupa *diphone* (gabungan dua buah fonem).

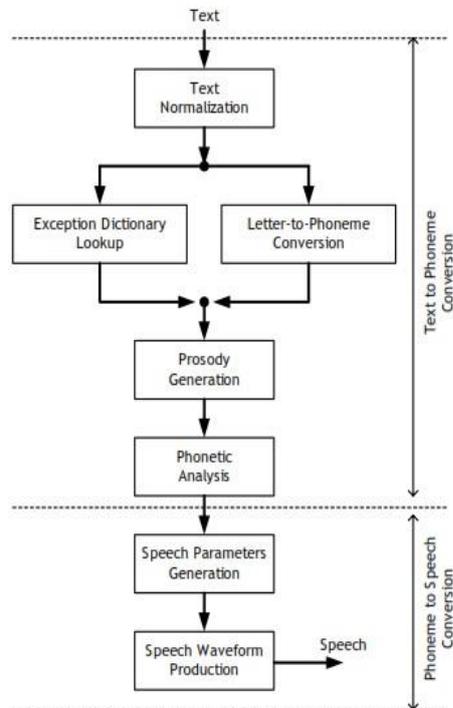
*Synthesizer* jenis ini dapat menghasilkan bunyi ucapan dengan tingkat kealamian (*naturalness*) yang tinggi.

Struktur sistem seperti di atas pada prinsipnya merupakan konfigurasi tipikal yang digunakan pada berbagai sistem *Text to Speech* berbagai bahasa. Namun demikian, pada setiap sub-sistem terdapat sifat-sifat serta proses-proses yang sangat spesifik dan sangat tergantung dari bahasanya.

Konversi dari teks ke fonem sangat dipengaruhi oleh aturan-aturan yang berlaku dalam suatu bahasa. Pada prinsipnya proses ini melakukan konversi dari simbol-simbol tekstual menjadi simbol-simbol fonetik yang merepresentasikan unit bunyi terkecil dalam suatu bahasa. Setiap bahasa memiliki aturan cara pembacaan dan cara pengucapan teks yang sangat spesifik. Hal ini menyebabkan implementasi unit konverter teks ke fonem menjadi sangat spesifik terhadap suatu bahasa. Untuk mendapatkan ucapan yang lebih alami, ucapan yang dihasilkan harus memiliki intonasi (*prosody*). Secara kuantisasi, prosodi adalah perubahan nilai *pitch* (frekuensi dasar) selama pengucapan kalimat dilakukan atau *pitch* sebagai fungsi waktu. Pada prakteknya, informasi pembentuk prosodi berupa data-data *pitch* serta durasi pengucapannya untuk setiap fonem yang dibangkitkan. Nilai-nilai yang dihasilkan diperoleh dari suatu model prosodi. Prosodi bersifat sangat spesifik untuk setiap bahasa, sehingga model yang diperlukan untuk membangkitkan data-data prosodi menjadi sangat spesifik juga untuk suatu bahasa. Beberapa model umum prosodi pernah dikembangkan, tetapi untuk digunakan pada suatu bahasa masih perlu banyak penyesuaian yang harus dilakukan.

Konverter fonem ke ucapan berfungsi untuk membangkitkan sinyal ucapan berdasarkan kode-kode fonem yang dihasilkan dari proses sebelumnya. Sub sistem ini harus memiliki pustaka setiap unit ucapan dari suatu bahasa. Pada sistem yang menggunakan teknik *diphone concatenation*, sistem harus didukung oleh suatu *diphone database* yang berisi rekaman segmen-segmen ucapan yang berupa *diphone*. Ucapan dalam suatu bahasa dibentuk dari satu set bunyi yang mungkin berbeda untuk setiap bahasa, oleh karena itu setiap bahasa harus dilengkapi dengan *diphone database* yang berbeda.

Tahapan-tahapan utama konversi dari teks menjadi ucapan dapat dinyatakan dengan diagram seperti terlihat pada Gambar 2.2 dibawah ini :



**Gambar 2.2** Urutan Proses Konversi dari Teks ke Ucapan

## 2.9 Huruf Braille

Huruf Braille adalah sejenis sistem tulisan sentuh yang digunakan oleh tunanetra. Sistem ini diciptakan oleh orang Perancis yang bernama Louis Braille. Ketika berusia 15 tahun, Braille membuat suatu tulisan untuk memudahkan tentara untuk membaca ketika gelap. Tulisan ini dinamakan huruf Braille. Namun ketika itu Braille tidak mempunyai huruf W Tahun 1834 ketika Louis Braille berusia awal 20-an, setelah bereksperimen dengan inovasinya itu selama lebih dari sepuluh tahun, sempurnalah sistem tulisan yang terdiri dari titik-titik timbul itu. Louis Braille hanya menggunakan enam titik —domino sebagai kerangka sistem tulisannya itu – tiga titik kebawah dan dua titik ke kanan seperti gambar 2.3 dibawah ini.



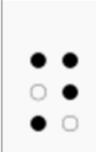
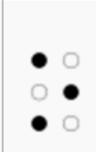
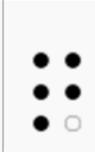
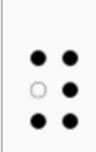
**Gambar 2.3** Kerangka Huruf Braille

Untuk memudahkan perujukan pada titik-titik dalam kerangka tersebut, masing-masing titik diberi nomor seperti gambar 2.4 dibawah ini :

1	4
2	5
3	6

**Gambar 2.4** Penomoran Kode *Braille*

Jadi dihitung mulai dari atas, titik-titik disebelah kiri diberi nomor 1, 2, dan 3, sedangkan titik-titik disebelah kanan diberi nomor 4, 5, dan 6. Penomoran ini akan mempermudah dalam belajar menulis dan membaca Braille. Berikut adalah huruf braille sesuai dengan abjad dan angka pada huruf biasa yang dapat dilihat pada gambar 2.5 dibawah ini.

									
A/1	B/2	C/3	D/4	E/5	F/6	G/7	H/8	I/9	J/0
									
K	L	M	N	O	P	Q	R	S	T
									
U	V	X	Y	Z					W

Gambar 2.5 Huruf Braille dan abjad.

### 2.10 Application Programming Interface(API)

*Application Programming Interface (API)* merupakan software interface yang terdiri atas kumpulan instruksi untuk mengintegrasikan dua aplikasi yang berbeda secara bersamaan. *API* bertujuan untuk mempercepat proses development dengan menyediakan function secara terpisah sehingga developer tidak perlu membuat fitur yang serupa. *API* terdiri dari berbagai elemen seperti function, protocols, dan tools lainnya yang memungkinkan developers untuk membuat aplikasi[7].

Secara struktural, *API* merupakan spesifikasi dari suatu *data structure, objects, functions*, beserta parameter-parameter yang diperlukan untuk mengakses *resource* dari aplikasi tersebut. Seluruh spesifikasi tersebut membentuk suatu *interface* yang dimiliki oleh aplikasi untuk berkomunikasi dengan aplikasi lain, dan *API* dapat digunakan dengan berbagai bahasa *programming*, ataupun hanya dengan menggunakan *URL (Uniform Resource Locator)* yang telah disediakan oleh suatu

*website*. API dapat diklasifikasikan menjadi beberapa kategori, hal ini dilihat dari abstraksi apa yang dideskripsikan di dalam sistem. Kategori-kategori ini diantaranya dapat dilihat pada tabel 2.1 dibawah ini :

**Tabel 2.1** Katagori *API*

<b>Deskripsi</b>	<b>Contoh</b>
API yang digunakan untuk fungsi dasar yang dapat dilakukan oleh komputer. Seperti proses I/O, eksekusi program.	API for MS Windows
API yang digunakan untuk memperluas kapabilitas dalam melakukan eksekusi terhadap suatu bahasa pemrograman.	Java API
API yang digunakan untuk mengakses data dan layanan yang disediakan dari suatu aplikasi.	API for mySAP ( <i>BAPI/Business Application Programming Interface</i> )
Digunakan untuk mengakses infrastruktur dari suatu komputer. Infrastruktur disini adalah komputer beserta <i>peripheral</i> seperti <i>storage</i> , aplikasi, dan lain-lain.	Amazon EC2 ( <i>Elastic Compute Cloud</i> ) untuk akses untuk <i>virtual computing</i> dan Amazon S3 ( <i>Simple Storage Service</i> ) untuk menyimpan data dalam jumlah besar.
API yang digunakan untuk mengakses <i>content</i> dan layanan yang disediakan oleh suatu <i>web application</i> .	<i>Facebook Graph API</i> yang digunakan untuk mengakses informasi yang dapat dibagikan.

### **2.11 Google API**

*Google* menyediakan berbagai *API (Application Programming Interface)* yang sangat berguna bagi pengembang aplikasi baik *web*, *mobile*, maupun *desktop* untuk memanfaatkan berbagai fitur yang disediakan oleh *Google* seperti: *AdSense*, *Search Engine*, *Translation*, *YouTube* dan lain sebagainya.

*API* secara sederhana bisa diartikan sebagai kode program yang merupakan antarmuka atau penghubung antara aplikasi atau web yang dibuat dengan fungsifungsi yang dikerjakan. *Google API* berarti kode program yang dapat ditambahkan pada aplikasi untuk memanfaatkan fungsi atau fitur yang disediakan *Google*. Contohnya developer dapat menambahkan fitur *Google Map* pada aplikasi yang sedang dibangun. *Google API* dapat dipelajari langsung melalui *Google Code*. Melalui *Google Code*, developer dapat belajar tentang *Google API* dan dapat mengimplementasikan pada aplikasi yang sedang dikembangkan.

## **2.12 Google Cloud Vision**

*Google Cloud Vision* dapat digunakan untuk mengidentifikasi objek-objek yang ada dalam sebuah gambar seperti teks, lambang, jenis benda produk secara digital. *Developer* dapat mengidentifikasi objek pada gambar, menganalisa emosi seseorang pada gambar, mendeteksi konten yang tidak pantas, dan mengekstraksi teks pada gambar dengan identifikasi bahasa otomatis.

Berikut adalah fitur yang dimiliki oleh *Google Cloud Vision* :

1) Label Detection

Mendeteksi katagori objek yang ada di gambar mulai dari kendaraan, binatang dll.

2) Explicit Content Detection

Mendeteksi apakah objek dalam gambar mengandung konten dewasa atau kekerasan.

3) Logo Detection

Mendeteksi logo produk terkenal dalam gambar.

4) Landmark Detection

Mendeteksi bangunan atau struktur terkenal baik buatan manusia ataupun alam di gambar.

5) Optical Character Recognition

Mendeteksi dan mengekstraksi teks pada gambar dengan berbagai jenis bahasa serta mendukung identifikasi bahasa otomatis.

6) Face Detection

Mendeteksi banyak wajah serta dapat mengenali emosi serta atribut pada wajah di gambar.

7) Image Attributes

Mendeteksi atribut umum pada gambar

8) Web Detection

Mencari gambar yang mirip diinternet seperti pada gambar.

### 2.13 Google Text To Speech

Penggunaan Google Text To Speech API memungkinkan developer untuk mengkonversi teks menjadi suara. Layanan ini dapat diproses secara offline dan online. Pada pemrosesan secara offline, bahasa yang didukung terbatas pada bahasa yang terdapat pada masing-masing device/smartphone. Berbeda dengan pemrosesan secara online yang memungkinkan dukungan bahasa yang lengkap.

Penggunaan secara online pengguna diharuskan memiliki koneksi internet untuk dapat mengakses database pada server Google secara online. *Google Text to Speech* mendukung 31 bahasa dalam proses *Text To Speech*. Bahasa yang didukung oleh *Google* dapat dilihat pada tabel 2.2 dibawah ini :

**Tabel 2.2** Bahasa yang di dukung *Google Text To Speech*

No	Nama Bahasa	Google Speak Code	Google Speak Code (asumsi)
1	<i>Arabic</i>	ar	
2	<i>Bulgarian</i>	-	ru
3	<i>Catalan</i>	ca	
4	<i>Chinese Simplified</i>	zh-CN	
5	<i>Chinese Traditional</i>	zh-TW	
6	<i>Czech</i>	cs	
7	<i>Danish</i>	da	
8	<i>Dutch</i>	nl	
9	<i>English</i>	en	
10	<i>Estonian</i>	-	nl
11	<i>Finnish</i>	fi	
12	<i>French</i>	fr	

13	<i>German</i>	de	
14	<i>Greek</i>	el	
15	<i>Haitian Creole</i>	ht	
16	<i>Hebrew</i>	-	iw
17	<i>Hindi</i>	hi	
18	<i>Hmong Daw</i>	-	nl
19	<i>Hungarian</i>	hu	
20	<i>Indonesian</i>	id	
21	<i>Italian</i>	it	
22	<i>Japanese</i>	ja	
23	<i>Korean</i>	ko	
24	<i>Latvian</i>	lv	
25	<i>Lithuanian</i>	-	nl
26	<i>Malay</i>	-	id
27	<i>Norwegian</i>	no	
28	<i>Persian</i>	-	ar
29	<i>Polish</i>	pl	
30	<i>Portuguese</i>	pt	
31	<i>Romanian</i>	ro	
32	<i>Russian</i>	ru	
33	<i>Slovak</i>	sk	
34	<i>Slovenian</i>	-	nl
35	<i>Spanish</i>	es	
36	<i>Swedish</i>	sv	
37	<i>Thai</i>	th	
38	<i>Turkish</i>	tr	
39	<i>Ukrainian</i>	-	ru
40	<i>Urdu</i>	-	ar
41	<i>Vietnamese</i>	vi	

## 2.14 Perangkat Lunak

Perangkat lunak (*software*) adalah program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain, dan cara penggunaan (*user manual*). Sebuah program komputer tanpa terasosiasi dengan dokumentasinya maka belum dapat disebut perangkat lunak (*software*).

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 sebuah sistem yang memiliki komponen berupa perangkat lunak yang memiliki hubungan satu sama lain untuk memenuhi kebutuhan pelanggan. Pelanggan adalah orang atau organisasi yang memesan atau membeli perangkat lunak (*software*) dari pengembang perangkat lunak atau bisa dianggap bahwa pelanggan adalah orang atau organisasi yang dengan sukarela mengeluarkan untuk memesan atau membeli perangkat lunak. User atau pemakai perangkat lunak adalah orang yang memiliki kepentingan untuk memakai atau menggunakan perangkat lunak untuk memudahkan pekerjaannya.

## **2.15 Basis Data**

Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan. Pada intinya basis data adalah media untuk penyimpanan data agar dapat diakses dengan mudah dan cepat.

Sistem informasi tidak dapat dipisahkan dengan kebutuhan akan basis data apapun bentuknya, entah berupa file teks ataupun Database Management System (DBMS). Kebutuhan basis data dalam sistem informasi meliputi memasukan, menyimpan, dan mengambil data serta membuat laporan berdasarkan data yang telah disimpan.

### **2.15.1 Databases Management System(DBMS)**

*DBMS (Database Management System)* atau dalam bahasa Indonesia sering disebut sebagai Sistem Manajemen Data adalah suatu sistem aplikasi yang digunakan untuk menyimpan, mengelola, dan menampilkan data. Suatu sistem aplikasi disebut DBMS jika memenuhi persyaratan minimal dapat menyediakan fasilitas untuk mengelola akses data, mampu menangani integrasi data, dan mampu menangani backup data.

DBMS sudah mulai berkembang sejak tahun 1960an. Kemudian sekitar tahun 1970an mulai berkembang teknologi Relational DBMS yaitu DBMS berbasis

relasional model. Relasional model pertama kali dikembangkan oleh Edgar J. Codd pada tahun 1970. Secara sederhana relasional model dapat dipahami sebagai suatu model yang memandang data sebagai sekumpulan tabel yang saling terkait.

Pada tahun 1980an mulai berkembang *Object Oriented DBMS (OODBMS)*. OODBMS berkembang seiring dengan perkembangan teknologi pemrograman berorientasi objek. Secara umum dapat diartikan bahwa OODBMS merupakan DBMS yang memandang data sebagai suatu objek.

Terdapat beberapa macam DBMS versi komersial yang paling sering digunakan di dunia saat ini, diantaranya *Oracle, Microsoft SQL Server, IBM DB2*, dan *Microsoft Access*. Sedangkan DBMS versi open source yang cukup berkembang dan sering digunakan saat ini adalah diantaranya *MySQL, PostgreSQL, Firebird*, dan *SQLite*. Hampir semua DBMS mengadopsi *SQL* sebagai bahasa untuk mengelola data pada DBMS.

### **2.15.2 Structured Query Language(SQL)**

*SQL (Structured Query Language)* adalah bahasa yang digunakan untuk mengelola data pada DBMS. Meskipun *SQL* diadopsi dan diacu sebagai bahasa standar oleh hampir sebagian besar *RDBMS* yang beredar saat ini, tetapi tidak semua standar yang tercantum dalam *SQL* diimplementasikan oleh seluruh DBMS tersebut. Sehingga terkadang terdapat perbedaan perilaku (hasil yang ditampilkan) oleh DBMS yang berbeda padahal query yang dimasukan sama

## **2.16 Pemograman Berbasis Objek**

Pendekatan berorientasi objek merupakan suatu teknik atau cara pendekatan dalam melihat permasalahan dan sistem (sistem perangkat lunak, sistem informasi, atau sistem lainnya). Pendekatan berorientasi objek akan memandang sistem yang akan dikembangkan sebagai suatu kumpulan objek yang berkorespondensi dengan objek-objek dunia nyata.

Ada banyak cara untuk mengabstraksikan dan memodelkan objek-objek tersebut, mulai dari abstraksi objek, kelas, hubungan antar kelas sampai abstraksi sistem. Saat mengabstraksikan dan memodelkan objek, data dan proses-proses yang dimiliki oleh objek akan dienkapsulasi (dibungkus) menjadi satu kesatuan.

Sistem berorientasi objek merupakan sebuah sistem yang komponennya dibungkus (dienkapsulasi) menjadi kelompok data dan fungsi. Setiap komponen dalam sistem tersebut dapat mewarisi atribut, sifat, dan komponen lainnya yang dapat berinteraksi satu sama lain.

Terdapat beberapa konsep utama pada metodologi berorientasi objek, diantaranya :

1. Kelas (class), kumpulan objek-objek dengan karakteristik yang sama. Kelas merupakan definisi statik dari himpunan objek yang sama yang mungkin lahir atau diciptakan dari kelas tersebut. Sebuah kelas akan mempunyai sifat (atribut), kelakuan (operasi/metode), hubungan (relationship), dan arti. Suatu kelas dapat diturunkan dari kelas yang lain, dimana atribut dari kelas semula dapat diwariskan ke kelas yang baru.
2. Objek (object), abstraksi dari sesuatu yang mewakili dunia nyata seperti benda, manusia, satuan organisasi, tempat, kejadian, struktur, status, atau hal-hal lain yang bersifat abstrak. Objek merupakan suatu entitas yang mampu menyimpan informasi (status) dan mempunyai operasi (kelakuan) yang dapat diterapkan atau dapat berpengaruh pada status objeknya. Objek mempunyai siklus hidup yaitu diciptakan, dimanipulasi, dan dihancurkan.
3. Abstraksi (abstraction), prinsip untuk merepresentasikan dunia nyata yang kompleks menjadi suatu bentuk model yang sederhana dengan mengabaikan aspek-aspek lain yang tidak sesuai dengan permasalahan.
4. Enkapsulasi (encapsulation), pembungkusan atribut data dan layanan (operasi-operasi) yang dimiliki objek untuk menyembunyikan implementasi dari objek sehingga objek lain tidak mengetahui cara kerjanya.
5. Pewarisan (inheritance), mekanisme yang memungkinkan suatu objek mewarisi sebagian atau seluruh definisi dan objek lain sebagai bagian dari dirinya.
6. Polimorfisme (polymorphism), kemampuan sesuatu objek untuk digunakan dibanyak tujuan yang berbeda dengan nama yang sama sehingga menghemat baris program.

## ***2.17 Unified Modeling Language(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. Unified Modelling Language (UML) juga merupakan sebuah bahasa yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem.

Terdapat beberapa diagram yang umum digunakan dalam pemodelan UML diantaranya :

1. Use Case Diagram, merupakan pemodelan untuk kelakuan (behavior) sistem yang akan dibuat. Use case mendeskripsikan sebuah interaksi antar satu atau lebih aktor dengan sistem yang akan dibuat. Secara kasar use case digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi itu.
2. Class Diagram, menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki atribut dan metode (operasi). Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas. Metode (operasi) fungsi-fungsi yang dimiliki oleh suatu kelas. Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki kelas main, kelas view, kelas controller, serta kelas model.
3. Sequence Diagram, menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui onjek-objek yang terlibat dalam sebuah use case beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada use case.

4. Activity Diagram, menggambarkan aliran kerja (workflow) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, melainkan aktivitas yang dapat dilakukan oleh sistem.
5. Component Diagram, dibuat untuk menunjukkan organisasi dan ketergantungan diantara kumpulan komponen dalam sebuah sistem. Diagram komponen fokus pada komponen sistem yang dibutuhkan dan ada di dalam sistem. Komponen dasar yang biasanya ada dalam suatu sistem adalah komponen user interface yang menangani tampilan, komponen bussiness processing yang menangani fungsi-fungsi proses bisnis, komponen data yang menangani manipulasi data, dan komponen security yang menangani keamanan sistem. Komponen lebih terfokus pada penggolongan secara umum fungsi-fungsi yang diperlukan.

### ***2.18 Android***

Android adalah sebuah sistem operasi untuk perangkat mobile berbasis Linux yang mencakup sistem operasi, middleware dan aplikasi. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Awalnya, Google Inc. membeli Android Inc., yang merupakan pendatang baru yang membuat peranti lunak untuk ponsel/smartphone. Kemudian untuk mengembangkan Android, dibentuklah Open Handset Alliance, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.

Pada saat perilis perdana Android, 5 November 2007, Android bersama Open Handset Alliance menyatakan mendukung pengembangan open source pada perangkat mobile. Di lain pihak, Google merilis kode-kode Android di bawah lisensi Apache, sebuah lisensi perangkat lunak dan open platform perangkat seluler.

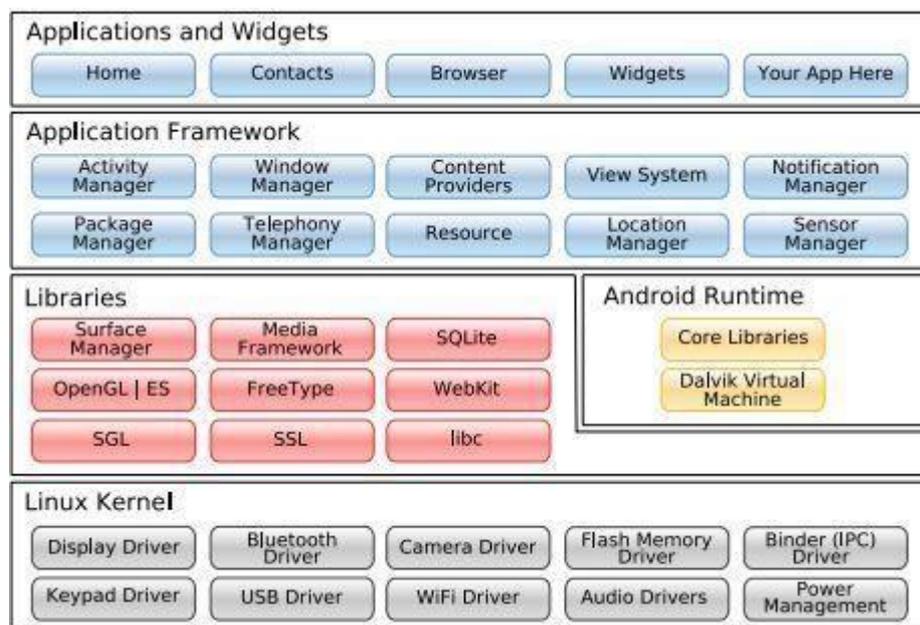
Di dunia ini terdapat dua jenis distributor sistem operasi Android. Pertama yang mendapat dukungan penuh dari Google atau Google Mail Services (GMS) dan kedua adalah yang benar-benar bebas distribusinya tanpa dukungan langsung

Google atau dikenal sebagai Open Handset Distribution (OHD). Tidak hanya menjadi sistem operasi di smartphone, saat ini Android menjadi pesaing utama dari Apple pada sistem operasi Tablet PC.

Pengembang memiliki beberapa pilihan ketika membuat aplikasi yang berbasis android. Sebagian besar pengembang menggunakan Eclipse yang tersedia secara bebas untuk merancang dan mengembangkan Aplikasi Android. Eclipse adalah IDE yang paling populer untuk pengembangan Android, karena memiliki Android plug-in yang tersedia untuk memfasilitasi pengembangan Android. Selain itu, Eclipse juga mendapat dukungan langsung dari Google untuk menjadi IDE pengembangan aplikasi Android, ini terbukti dengan adanya penambahan plugin untuk eclipse untuk membuat project android dimana source software langsung dari situs resmi Google. Akan tetapi, hal tersebut tidak menutup kemungkinan untuk menggunakan IDE yang lain seperti Netbeans untuk melakukan pengembangan android

### 2.18.1 Arsitektur Android

Secara garis besar Arsitektur Android dapat dijelaskan dan dilihat pada gambar 2.6 dibawah ini:



**Gambar 2.6** Arsitektur Android.

Berikut ini adalah penjelasan dari gambar 2.6 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 *contont-providers* yang berupa sms dan lain sebagainya. Komponen-komponen yang termasuk didalam *Applications Frameworks* diantaranya *Views*, *Content Provider*, *Resouce 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*.

### **2.19 Java Development Kit(JDK)**

JDK (Java Development Kit) adalah sebuah perangkat peralatan yang digunakan untuk membangun perangkat lunak dengan menggunakan bahasa pemrograman Java. JDK berjalan diatas sebuah virtual machine yang dinamakan JVM (Java Virtual Machine).

Dokumentasi JDK berisi spesifikasi API, deskripsi fitur, panduan pengembang, referensi halaman untuk perkakas JDK dan utilitas, demo, dan link ke informasi terkait. Dokumentasi ini juga tersedia dalam bundel yang dapat diunduh dan diinstal pada komputer.

## ***2.20 Netbeans***

NetBeans awalnya dibangun pada tahun 1996 sebagai Xelfi (untuk pemrograman Delphi) oleh seorang mahasiswa dari Charles University di Paraguay. Pada tahun 1997, Roman Stanek membangun sebuah perusahaan dan merilis versi komersial dari NetBeans hingga akhirnya dibeli oleh Sun Microsystems pada 1999. Hingga saat ini platform NetBeans telah banyak berkembang di bawah SunMicrosystem[8].

NetBeans merupakan platform framework dan IDE (integrated development environment) yang digunakan untuk pengembangan aplikasi desktop yang menggunakan Bahasa Java, dan beberapa bahasa lain, seperti Groovy, C, C++ dan banyak lagi. NetBeans IDE dibangun menggunakan Bahasa Java dan dapat dijalankan pada Windows, OS X, Linux, Solaris dan sistem operasi lain yang mendukung JVM. IDE NetBeans merupakan alat pengembangan aplikasi yang terintegrasi.

NetBeans IDE mendukung pengembangan Program yang menggunakan Bahasa Java dari semua versi (Java SE, Java ME, Java EE). Platform Netbeans memperbolehkan pembangunan aplikasi dengan menggunakan modul-modul. Aplikasi yang dibangun menggunakan NetBeans dapat dikembangkan oleh pihak ketiga. Platform NetBeans merupakan platform yang dapat digunakan ulang (reusable) untuk mempermudah pembangunan program menggunakan Bahasa Java.

## ***2.21 Android Software Development Kit(SDK)***

Android SDK adalah tools API (Application Programming Interface) yang diperlukan untuk memulai mengembangkan aplikasi pada platform Android menggunakan bahasa pemrograman Java. Android merupakan subset perangkat lunak untuk ponsel yang meliputi sistem operasi, middleware, dan aplikasi kunci

yang di-release oleh Google. Saat ini disediakan Android SDK sebagai alat bantu dan API untuk mulai mengembangkan aplikasi pada platform Android menggunakan bahasa pemrograman Java. 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).
- 9) Lingkungan Development yang lengkap dan kaya termasuk perangkat emulator, tools untuk debugging, profil, dan kinerja memori.

## **2.22 Android Studio**

Android studio merupakan sebuah software tools *Integrated Development Environment* (IDE) untuk platform Android. Android studio ini diluncurkan pada tanggal 16 Mei 2013 pada Konferensi Google I/O oleh Produk Manajer Google, Ellie Powers[9].

Android studio ini bersifat *free* dibawah Apache License 2.0. Android Studio awalnya dimulai dengan versi 0.1 pada bulan Mei 2013, kemudian dibuat versi beta 0.8 yang dirilis pada bulan Juli 2014, yang paling terbaru dirilis v.3 pada bulan Oktober 2017. Berbasiskan JetBrains' IntelliJ IDEA, Studio didesain khusus untuk Android Development. Sekarang ini sudah bisa didownload untuk Windows, Mac OS X, dan Linux[9].

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 :

- 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
- 9) Dukungan bawaan untuk Google Cloud Platform, mempermudah pengintegrasian Google Cloud Messaging dan App Engine[10]

**Syarat Hardware:**

## Windows :

1. Microsoft Windows 7/8/10 (32 atau 64 bit)
2. RAM minimum 3 GB, RAM yang digunakan disarankan 8 GB; ditambah 1 GB untuk Emulator Android
3. Ruang Disk minimum yang tersedia 2 GB. Disarankan 4 GB (500 MB untuk IDE + 1,5 GB untuk Android SDK dan gambar sistem emulator)
4. Resolusi layar minimum 1280 x 800

## Mac :

1. Mac OS X 10.10 (Yosemite) atau lebih tinggi, hingga 10.12 (mac OS Sierra).
2. RAM minimum 3 GB, RAM yang disarankan 8 GB; ditambah 1 GB untuk Emulator Android
3. Ruang Disk minimum yang tersedia 2 GB. Disarankan 4 GB (500 MB untuk IDE + 1,5 GB untuk Android SDK dan gambar sistem emulator)
4. Resolusi layar minimum 1280 x 800

## Linux :

1. Desktop GNOME atau KDE. Telah diuji pada Ubuntu 12.04, Precise Pangolin (distribusi 64-bit dan 32-bit).
2. Distribusi 64-bit yang mampu menjalankan aplikasi 32-bit.
3. GNU C Library (glibc) 2.19 atau yang lebih baru.
4. RAM minimum 3 GB, RAM yang disarankan 8 GB; ditambah 1 GB untuk Emulator Android.
5. Ruang Disk minimum yang tersedia 2 GB. Disarankan 4 GB (500 MB untuk IDE + 1,5 GB untuk Android SDK dan gambar sistem emulator).
6. Resolusi layar minimum 1280 x 800.[9]

