

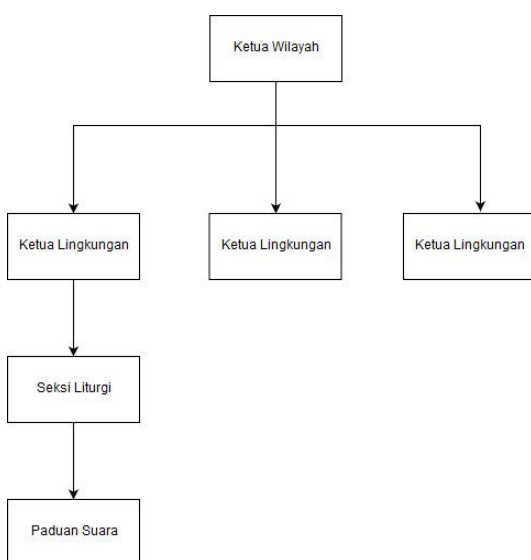
## BAB 2

### TINJAUAN PUSTAKA

#### 2.1 Profil Tempat Penelitian

Gedung Serba Guna (GSG) Wilayah Kebon Kangkung seluas sekitar 160 m<sup>2</sup> terletak di lahan seluas 4865 m<sup>2</sup>, dibangun tahun 1987. Selain GSG, di lahan ini juga terdapat SD dan SMP Slamet Riyadi. GSG yang terletak di Jalan Kebon Kangkung X ini digunakan sebagai sarana kegiatan seperti Rukun Ibu-ibu Katolik (Rikat), Ibadat Sabda, dan perayaan Ekaristi. Kemudian Gedung Serba Guna (GSG) tersebut dijadikan menjadi gereja St. Stefanus. Dalam melakukan kegiatan keagamaan.

Gereja tersebut memiliki 5 lingkungan yang terdiri dari lingkungan Sukasari, Sukapura, Kiara, Binong dan Sariwates. Dalam . Gereja memiliki pemimpin wilayah yang akan bertanggung jawab atas kegiatan yang belangsung di gereja kemudian dibawahnya diikuti oleh ketua lingkungan dan seksi-seksi. Berikut adalah Gambar 2.1 yang akan menunjukkan struktur organisasi yang terdapat pada gereja St. Stefanus Kebon Kangkung :



Gambar 2.1 Struktur Organisasi

Berikut adalah uraian tugas dari setiap bagian dari struktur di atas :

- a. Ketua Wilayah : Bertanggung jawab atas seluruh kegiatan keagamaan didalam mau diluar gereja
- b. Ketua Lingkungan : Bertanggung jawab dalam mengurus lingkungan cakupan gereja
- c. Seksi Liturgi : Bertanggung jawab dalam pelaksanaan misa (kegiatan keagamaan orang katolik)
- d. Paduan Suara : Bertanggung jawab membantu jalannya misa dengan cara bernyanyi

## 2.2 Logo Tempat penelitian

Berikut adalah logo dari Gereja ST. Stefanus pada Gambar 2.2 :



Gambar 2.2 Logo Gereja

## 2.3 Multimedia

Multimedia merupakan kombinasi dari teks, gambar, seni grafik, suara, animasi dan elemen-elemen video yang dimanipulasi secara digital. Tampilan dan cita rasa dari sebuah proyek multimedia harus menyenangkan, estetis, mengundang dan mengikat. Proyek harus memuat konsistensi audio dan visual, hanya dengan menggunakan seluruh element yang mendukung pesan dari program [5].

## 2.4 Pengertian Musik

Pada bagian ini akan dibahas pengertian dari musik dan bagian-bagian penting yang terdapat didalam musik. Musik dalam bahasa Yunani disebut dengan *Muse*. *Muse* itu sendiri adalah sebutan untuk dewa-dewi mitologi Yunani yang bertanggung jawab terhadap perkembangan seni dan ilmu pengetahuan dalam Kamus Besar Bahasa Indonesia Musik adalah “ ilmu atau seni menyusun nada atau suara dalam urutan, kombinasi, dan hubungan temporal untuk menghasilkan komposisi (suara) yang mempunyai kesatuan dan kesinambungan atau nada atau suara yang disusun demikian rupa sehingga mengandung irama, lagu, dan keharmonisan (terutama yang menggunakan alat-alat yang dapat menghasilkan bunyi-bunyi itu)” [6].

## 2.5 Musik Vokal

Musik vokal adalah musik yang bersumber dari suara manusia, yang bisa dimainkan oleh seorang penyanyi atau sekelompok orang. Jika dinyanyikan perorangan disebut solo, dan jika dinyanyikan secara berkelompok disebut suara bersama (*samen zingen*). Suara bersama ini apabila dinyanyikan dengan harmoni dan berbagai warna suara (timbre) disebut paduan suara atau *choir* (koor) [7].

## 2.6 Paduan Suara

Paduan Suara adalah musik yang bersumber dari suara manusia yang dinyanyikan secara bersama dengan harmonis dan berbagai warna suara (timbre) seperti sopran, meso sopran, alto, kontraalot, tenor, bariton, dan bass.

Paduan suara atau kor (dari bahasa Belanda, *koor*) merupakan istilah yang merujuk kepada ensemble musik yang terdiri atas penyanyi-penyanyi maupun musik yang dibawakan oleh ensemble tersebut. Umumnya suatu kelompok paduan suara membawakan musik paduan suara yang terdiri atas beberapa bagian suara (bahasa Inggris: *part*, bahasa Jerman: *Stimme*). Dalam perkembangannya, pada tahun 800-an suatu jenis musik baru yang disebut musik polyphonic berkembang di Eropa. Dalam musik polyphonic ini beberapa melodi dimainkan atau dinyanyikan dalam waktu yang bersamaan. Pada akhir tahun-tahun 1100-an,

karya-karya musik yang ditulis oleh beberapa komponis, seperti komponis Perancis Perotin menggabungkan semua unsur musik, seperti melodi, irama, harmoni dan *polyphonic* dan karya-karya tersebut ditampilkan oleh paduan suara, penyanyi solo dengan iringan berbagai instrumen musik. Sebuah karya musik paduan suara yang terkenal pada tahun 1300-an adalah Misa Notre Dame, yang digubah oleh komponis dan penyair Perancis *Guillaume de Machaut* pada tahun 1364.

Pengertian paduan suara adalah Penyajian musik *vocal* yang terdiri dari 15 orang atau lebih yang memadukan berbagai warna suara menjadi satu kesatuan yang utuh dan dapat menampilkan jiwa lagu yang dibawakan. Paduan suara biasanya dipimpin oleh seorang dirigen atau choirmaster yang umumnya sekaligus adalah pelatih paduan suara tersebut [7]. Umumnya paduan suara terdiri atas empat bagian suara (misalnya sopran, alto, tenor, dan bas), walaupun dapat dikatakan bahwa tidak ada batasan jumlah suara yang terdapat dalam paduan suara. Selain empat suara, jumlah jenis suara yang paling lazim dalam paduan suara adalah tiga, lima, enam, dan delapan. Bila menyanyi dengan satu suara, paduan suara tersebut diistilahkan menyanyi secara unisono. Paduan suara dapat bernyanyi dengan atau tanpa iringan alat musik. Bernyanyi tanpa iringan alat musik biasanya disebut sebagai bernyanyi *acappella*. Bila bernyanyi dengan iringan, alat musik pengiring paduan suara dapat terdiri atas alat musik apa saja, satu, beberapa, atau bahkan suatu orkestra penuh. Untuk latihan paduan suara, alat pengiring yang digunakan biasanya adalah piano, termasuk bahkan jika pada penampilannya digunakan alat musik lain atau ditampilkan secara *acappella*.

a. Struktur Paduan Suara

Pada umumnya paduan suara terdiri atas empat bagian suara (misalnya sopran, alto, tenor, dan bass), walaupun dapat dikatakan bahwa tidak ada batasan jumlah suara yang terdapat dalam paduan suara. Selain empat suara, jumlah jenis suara yang paling lazim dalam paduan suara adalah tiga, lima, enam, dan delapan. Bila menyanyi dengan satu suara, itu dinamakan atau diistilahkan secara unison untuk latihan paduan suara, alat pengiring yang digunakan biasanya adalah piano, keyboard, pianika, gitar, biola. Sedangkan tanpa iringan biasanya ditampilkan secara *acapella*.

b. Jenis-Jenis Paduan Suara

1. Paduan Suara UNISONO yaitu Paduan suara dengan menggunakan satu suara.
2. Paduan Suara 2 suara sejenis, yaitu paduan suara yang menggunakan 2 suaramanusia yang sejenis, contoh : Suara sejenis Wanita, Suara sejenis Pria, Suaras sejenis anak-anak.
3. Paduan Suara 3 sejenis S – S – A, yaitu paduan suara sejenis dengan menggunakan suara Sopran 1, Sopran 2, dan Alto.
4. Paduan Suara 3 suara Campuran S – A – B, yaitu paduan suara yang menggunakan 3 suara campuran , contoh : Sopran, Alto Bass.
5. Paduan suara 3 sejenis T- T – B, yaitu paduan suara 3 suara sejenis pria dengan suara Tenor 1, Tenor 2, Bass.
6. Paduan Suara 4 suara Campuran, yaitu paduan suara yang menggunakan suara campuran pria dan wanita, dengan suara S – A – T – B. Sopran, Alto, Tenor, Bass.

c. Hal-hal yang harus diperhatikan dalam Paduan Suara

Berikut ini adalah unsur-unsur yang perlu diperhatikan di dalam Paduan Suara diantaranya adalah :

1. Artikulasi adalah cara pengucapan kata demi kata yang baik dan jelas dalam bernyanyi.
2. Pernafasan adalah usaha untuk menghirup udara sebanyak-banyaknya, kemudian disimpan, dan dikeluarkan sedikit demi sedikit sesuai dengan keperluan. Pernafasan di bagi tiga jenis, yaitu :
  - a. Pernafasan Dada pernapasan yang cocok untuk nada-nada rendah, penyanyi mudah lelah.
  - b. Pernafasan Perut adalah pernapasan yang membuat udara cepat habis, kurang cocok digunakan dalam menyanyi, karena akan cepat lelah.
  - c. Pernafasan Diafragma adalah pernafasan yang paling cocok digunakan untuk menyanyi, karena udara yang digunakan akan

mudah diatur pemakaiannya, mempunyai *power* dan stabilitas *vocal* yang baik.

3. *Phrasing* adalah aturan pemenggalan kalimat yang baik dan benar sehingga mudah dimengerti dan sesuai dengan kaidah-kaidah yang berlaku.
4. Sikap Badan adalah posisi badan ketika seseorang sedang nyanyi, bisa sambil duduk, atau berdiri, yang penting saluran pernafasan jangan sampai terganggu.
5. Resonansi adalah usaha untuk memperindah suara dengan mefungsikan rongga-rongga udara yang turut bervibrasi/ bergetar disekitar mulut dan tenggorokan.
6. Vibrato adalah Usaha untuk memperindah sebuah lagu dengan cara memberi gelombang/ suara yang bergetar teratur, biasanya di terapkan di setiap akhir sebuah kalimat lagu.
7. Improvisasi adalah usaha memperindah lagu dengan merubah/menambah sebagian melodi lagu dengan profesional, tanpa merubah melodi pokoknya.
8. Intonasi adalah tinggi rendahnya suatu nada yang harus dijangkau dengan tepat.

## **2.7 Notasi Lagu**

Sistem penulisan ekspresi musik yang di tuangkan dalam bentuk simbol yang berwujud angka atau gambar not. Terdapat dua notasi lagu yang terdapat yaitu notasi balok dan notasi angka, berikut adalah penjelasan tentang kedua notasi lagu tersebut:

### **2.7.1 Not Balok**

Dalam notasi balok, sistem paranada bergaris lima digunakan sebagai dasar. Bersama dengan keterangan mengenai tempo, ketukan, dinamika, dan instrumentasi yang digunakan, not ditempatkan pada paranada dan dibaca dari kiri ke kanan. Durasi nada dilambangkan dengan nilai not yang berbeda-beda,

sedangkan tinggi nada dilambangkan dalam posisi not secara vertikal pada paranada.



Gambar 2. 3 Not Balok

### 2.7.2 Not Angka

Not angka merupakan notasi lagu yang direpresentasikan menggunakan angka. Berikut adalah tabel lengkap bagaimana suatu notasi dituliskan kedalam bentuk angka :

Tabel 2. 1 Not Angka

Notasi Angka Yang Umum Digunakan			
Not	Baca	Not	Baca
1 = Do		1̇ = Di	
2 = Re		2̇ = Ri	
3 = Mi			
4 = Fa		4̇ = Fi	
5 = Sol		5̇ = Sel	
6 = La			
7 = Si		7̇ = Sa	

Dalam notasi angka, not ditentukan dengan angka 1 (do), 2 (re), 3 (mi), 4 (fa), 5 (sol), 6 (la) dan 7 (si). Nada 1 tanpa titik merupakan nada C natural di notasi balok. Tanda satu titik di atas not, menunjukkan bahwa not tersebut naik satu birama dari nada asli, sedangkan tanda satu titik di bawah not menunjukkan bahwa

not tersebut turun satu birama dari nada asli [8]. Cara Membaca Notasi Angka adalah sebagai berikut :

1. Do = G menunjukkan nada dasar lagu tersebut.
2. 4/4 menunjukkan Tanda birama yang menunjukkan ritme lagu. Angka di bagian atas tanda birama menunjukkan jumlah ketukan per birama, sedangkan angka di bawah menunjukkan nilai not per ketukan. Tanda birama 4/4 di sini menunjukkan bahwa terdapat empat ketukan dalam birama, satu ketukan kuat diikuti tiga ketukan lemah, dan masing-masing ketukan bernilai not seperempat
3. Tempo = 66 menunjukkan tempo lagu, artinya dalam satu menit ada 66 ketuk.
4. 4. SATB menunjukkan tipe suara yang menyanyikan baris tersebut.
5. P berarti 'piano' yang berarti lembut, artinya lagi ini dengan dinamika yang lembut.
6. Tanda *Crescendo* yang dilanjutkan dengan tanda *decrescendo*, menunjukkan ada perubahan dinamika, yakni mengeras, kemudian melembut lagi.
7. Garis birama yang merupakan pemisah antar birama.

Do = G; 4/4; Tempo = 66

p

S	5	.	3	1	4	4	5	6	5	.	3	.
A	5	.	3	1	1	4	5	6	1	.	1	.
T	1	.	5	3	6	6	7	1	1	.	5	.
B	1	.	2	3	4	1	2	3	1	.	1	.

Gambar 2. 4 Not Angka

### 2.7.3 Nada Dasar

Pada Sebuah partitur lagu ada juga yang dinamakan nada dasar. Nada dasar dijadikan sebagai patokan tinggi rendahnya suatu lagu dinyanyikan. Nada dasar adalah nada pertama dalam urutan tangga nada yang nada pertama yang dijadikan



sebagai dasar dalam menentukan susunan nada dalam sebuah tangga nada [9]. Nada dasar memiliki susunan interval tangga nada, yaitu sebagai berikut :  $1 - 1 - 1/2 - 1 - 1 - 1 - 1/2$ . Apapun nada dasar yang digunakan harus memenuhi susunan interval tangga nada yang ada.

## 2.8 Pengertian Citra(*image*)

Menurut *Oxford Dictionaries* ([oxforddictionaries.com](http://oxforddictionaries.com)), citra atau image adalah “sebuah representasi dari bentuk eksternal dari seseorang atau sesuatu, sesuatu yang dapat dihasilkan dari kamera, teleskop, mikroskop, atau perangkat lain, yang dapat ditampilkan pada layar komputer atau video, atau penampilan optik yang dihasilkan oleh cahaya dari suatu objek tercermin dalam cermin atau yang dibiaskan melalui lensa” [10].

### 2.8.1 Pengolahan Citra

Pengolahan citra digital adalah metode yang digunakan untuk mengolah citra pada komputer sehingga sesuai dengan citra yang dibutuhkan. Secara umum, operasi pengolahan citra dapat diklasifikasikan dalam beberapa jenis sebagai berikut [10] :

a. *Image enhancement*

Operasi ini bertujuan untuk memperbaiki kualitas citra. Dengan operasi ini, ciri khusus pada citra dapat lebih ditonjolkan.

b. *Image restoration*

Operasi ini bertujuan untuk melakukan perbaikan pada citra dengan cara meminimumkan atau menghilangkan cacat pada citra.

c. *Image compression*

Operasi ini bertujuan untuk menghemat memori yang diperlukan untuk penyimpanan citra. Hal penting yang harus diperhatikan dalam kompresi citra ini adalah citra yang dikompresi harus tetap memiliki kualitas gambar yang baik.

d. *Image segmentation*

Operasi ini bertujuan untuk memecah suatu citra ke dalam beberapa segmen dengan suatu kriteria tertentu.

e. *Image analysis*

Operasi ini bertujuan untuk menghitung besaran kuantitatif dari citra untuk menghasilkan deskripsinya. Teknik ini mengekstraksi ciri tertentu yang membantu mengidentifikasi objek.

f. *Image reconstruction*

Operasi ini bertujuan untuk membentuk ulang objek dari beberapa citra hasil proyeksi.

### **2.8.2 Optical Character Recognition**

*Optical character recognition* (OCR) adalah proses konversi gambar huruf menjadi karakter ASCII yang dikenali oleh komputer. Gambar huruf yang dimaksud dapat berupa hasil scan dokumen, hasil print-screen halaman web, hasil foto, dan lain-lain [11]. File citra yang akan dibaca di kirim ke komputer atau perangkat dengan OCR dan diproses menjadi teks. Berikut ini merupakan cara kerja OCR:

a. Meratakan gambar (*Auto Deskewing*)

Citra akan diatur kemiringannya, apabila hasil scan ada kemiringan, maka gambar akan dibuat sejajar atau lurus.

b. Menganalisa (*Analysis*)

OCR akan menganalisa dan memisahkan bagian teks dan bagian gambar.

c. Otomatis mengatur arah gambar (*Auto Orientation*)

OCR akan mengambil sebagian area pada citra dan mengidentifikasi arah teks yang benar. Citra akan diputar ke arah yang benar, pilihannya adalah  $90^\circ$ ,  $180^\circ$  atau  $270^\circ$ .

d. Memisahkan setiap karakter huruf dan angka (*Separating single character*) OCR akan memisahkan setiap karakter yang terdapat pada citra menjadi sebuah huruf atau angka.

e. Mengidentifikasi gambar (*Capturing the features of the characters and comparing*)

OCR akan mengidentifikasi satu persatu setiap gambar yang sudah dipisahkan, dan melakukan pengecekan terhadap database yang dimiliki oleh OCR, dan menetapkan huruf atau angka yang akan digunakan.

f. Menghasilkan file akhir (*Recognition result output*)

Setelah setiap citra sudah dirubah menjadi huruf atau angka (dalam format teks) maka OCR akan menghasilkan sebuah file dengan format teks, seperti Microsoft Word, Microsoft Excel, TXT, Rich Text dan Searchable PDF, tergantung dari pengaturan awal. Metode-metode yang ada pada teknologi OCR yaitu, *matrix matching* dan *feature extraction* [12]. Berikut adalah penjelasan lebih lanjut mengenai *matrix matching* dan *feature extraction*.

1. *Matrix Matching*

Metode pertama dan yang paling simpel adalah *matrix matching*. Lebih dikenal pula sebagai pattern matching, karena scanner akan mendeteksi karakter yang dipindai lalu menyamakan dengan database dari pola atau outline dari setiap karakter yang dianalisa dari bentuk citra. Maka ketika scanner memindai sebuah karakter, sistem akan mendeteksi bentuk karakter tersebut dan menyamakan bentuk, sudut, dan pola karakter dari citra dengan yang ada di database.

Jika karakter cocok maka karakter yang dipindai tersebut akan diidentifikasi sebagai karakter yang sama dalam database. Hanya saja teknik ini memiliki keterbatasan, karena database ini memiliki keterbatasan akan jumlah variasi bentuk karakter. Font saat ini pun beraneka ragam, pola dan bentuknya semakin kompleks. Metode ini merupakan metode yang paling banyak digunakan pada OCR saat ini.

2. *Feature Extraction*

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 sistem. Metode ini akan mendeteksi identitas sebuah karakter dari ruang kosong, bentuk yang berdekatan, garis diagonal, perpotongan dan sebagainya. Maka proses pengidentifikasiannya akan lebih kompleks.

*Feature Extraction* biasanya digunakan untuk mendeteksi karakter yang bentuknya tak standar, seperti tulisan tangan atau huruf-huruf ber-ornamen. *Matrix Matching* lebih cocok untuk dokumen yang menggunakan jenis huruf yang umum, seperti Times New Romans, Arial, atau font yang memiliki bentuk dasar karakter huruf lainnya. Database karakter pada *Feature Extraction* biasanya meliputi rangkaian dasar karakter angka dari 0-9 dan huruf a-z

## 2.9 Midi

*Musical Instrument Digital Interface* (MIDI) adalah teknologi yang memungkinkan alat musik elektronik untuk berkomunikasi satu sama lain, dengan mengirim dan menerima informasi kinerja [13]. Adapun struktur penyimpanan *file* dalam *MIDI* secara umum adalah sebagai berikut :

### a. *Chunks*

File MIDI disusun menjadi potongan. Setiap potongan terdiri dari:

1. A 4-byte chunk type (ascii).
2. A 4-byte length (32 bits, msb first).
3. length bytes of data.

terdapat 2 jenis chunk yaitu :

1. *Chunks* Header yang memiliki jenis chunk "MThd".
2. Lacak *Chunks* yang memiliki jenis chunk "MTrk".

### b. Value Representations

Di dalam file MIDI, ada berbagai informasi selain data dasar MIDI, seperti *delta-times* dan *meta-events*. Di mana informasi tersebut mengandung nilai numerik, ini umumnya diwakili dalam salah satu dari dua format:

1. *Binary*.

## 2. *Variable length quantity.*

Kedua hal tersebut merupakan struktur utama yang terdapat pada *midi file* [13].

### 2.10 Unified Modeling Language (UML)

Pada pertengahan 1990-an, metodologi yang paling terkenal adalah yang ditemukan oleh Ivar Jacobson, James dan Grady Booch. Masing-masing memiliki perusahaan konsultan sendiri, menggunakan metodologi sendiri dan notasi sendiri. Pada tahun 1996, Jacobson dan Rumbaugh bergabung Rasional Corporation (didirikan oleh Booch), dan mereka mengembangkan satu set notasi yang dikenal sebagai *Unified Modeling Language* (UML) [15]. UML memiliki 13 jenis diagram. Spesifikasi UML tidak mengatakan di mana diagram harus digunakan dalam metodologi tertentu :

- a. *Use Case Diagram* mengkategorikan cara di mana sistem yang digunakan.
- b. *Class Diagram* menunjukkan kelas dan bagaimana mereka dapat dipasang bersama-sama (mereka juga dapat menunjukkan object).
- c. *Object Diagram* memperlihatkan obyek saja dan bagaimana mereka dapat dipasang bersama-sama.
- d. *Activity Diagram* menunjukkan aktivitas oleh manusia atau objek dalam cara yang mirip dengan Flow Chart.
- e. *State Machine Diagram* menunjukkan berbagai state setiap obyek dengan siklus hidup yang menarik atau rumit.
- f. *Communication Diagram* menunjukkan pesan yang dikirim antara objek dalam skenario tertentu.
- g. *Sequence Diagram* menunjukkan informasi yang sama dengan communication diagram, tetapi menekankan urutan daripada koneksi.
- h. *Package Diagram* menunjukkan bagaimana kelas terkait dikelompokkan bersama-sama, untuk kepentingan pengembang.
- i. *Deployment Diagram* menunjukkan mesin, proses dan artefak dikerahkan untuk jadi seorang sistem.
- j. *Component Diagram* menunjukkan komponen dapat digunakan kembali (benda atau subsistem) dan interface mereka.

- k. *Interaction Overview Diagram* menunjukkan langkah-langkah individu dari suatu kegiatan menggunakan sequence diagram.
- l. *Timing Diagram* menunjukkan kendala waktu yang tepat untuk pesan dan menyatakan objek.
- m. *Composite Structure Diagram* menunjukkan bagaimana benda cocok bersama dalam suatu agregasi atau komposisi, menampilkan antarmuka dan berkolaborasi benda.

## 2.11 Android

Aplikasi Android adalah aplikasi yang ditulis dalam bahasa pemrograman Java. Android SDK Tools mengompilasi kode bersamaan data dan file sumber daya dan menjadikannya sebuah aplikasi: sebuah paket aplikasi Android memiliki ekstensi file yang bernama `.apk`. Satu file `.apk` berisi semua materi aplikasi Android dan merupakan tipe file yang akan digunakan pada perangkat Android untuk memasang aplikasi [16]. Setelah dipasang di perangkat, setiap aplikasi Android tinggal di kotak pasir keamanannya sendiri:

- a. Sistem operasi Android merupakan sistem Linux multi-pengguna yang di dalamnya setiap aplikasi adalah pengguna berbeda.
- b. Secara default, sistem menetapkan ID pengguna Linux unik kepada setiap aplikasi (ID ini hanya digunakan oleh sistem dan tidak diketahui aplikasi). Sistem menetapkan izin bagi semua file dalam aplikasi sehingga hanya ID pengguna yang diizinkan yang bisa mengaksesnya.
- c. Setiap proses memiliki mesin virtual (VM) sendiri, sehingga kode aplikasi yang berjalan secara terisolasi dari aplikasi lainnya.
- d. Secara default, setiap aplikasi berjalan dalam proses Linux-nya sendiri. Android memulai proses bila ada komponen aplikasi yang perlu dijalankan, kemudian mematikan proses bila tidak lagi diperlukan atau bila sistem harus memulihkan memori untuk digunakan aplikasi lain.

Dengan cara ini, sistem Android mengimplementasikan *prinsip privilese minim*. Ini berarti, secara default aplikasi hanya memiliki akses ke komponen yang diperlukannya untuk melakukan pekerjaannya dan tidak lebih dari itu. Hal ini

menghasilkan lingkungan yang sangat aman sehingga aplikasi tidak bisa mengakses bagian sistem bila tidak diberi izin. Akan tetapi, ada beberapa cara bagi aplikasi untuk berbagi data dengan aplikasi lain dan bagi aplikasi untuk mengakses layanan sistem:

- a. Dua aplikasi bisa diatur untuk menggunakan ID pengguna Linux yang sama, dalam hal ini keduanya bisa saling mengakses file masing-masing. Untuk menghemat sumber daya sistem, aplikasi dengan ID pengguna yang sama juga bisa diatur agar berjalan dalam proses Linux yang sama dan menggunakan VM yang sama ( aplikasi juga harus ditandatangani dengan sertifikat yang sama).
- b. Aplikasi bisa meminta izin akses ke data perangkat seperti kontak pengguna, pesan SMS, storage lepas-pasang (kartu SD), kamera, Bluetooth, dan lainnya. Pengguna secara eksplisit harus memberikan izin ini. Untuk informasi selengkapnya, lihat Bekerja dengan Izin Sistem. Hal tersebut mencakup dasar-dasar tentang cara aplikasi Android berada di dalam sistem. Bagian dokumen selanjutnya memperkenalkan Anda pada:
  1. Komponen kerangka kerja inti yang mendefinisikan aplikasi.
  2. File manifes tempat Anda mendeklarasikan komponen dan fitur yang diperlukan perangkat untuk aplikasi.
  3. Sumber daya yang terpisah dari kode aplikasi dan memungkinkan aplikasi mengoptimalkan perilakunya untuk beragam konfigurasi perangkat.

## 2.12 Tools Bantuan

Berikut adalah *tools* yang akan digunakan untuk membantu dalam Pembangunan sistem pengenalan not angka ini dan pembuatan partitur yang meliputi bahasa pemrograman, IDE, *library* yang digunakan :

### 2.12.1 Tess-two

Sebuah API untuk Android yang menambahkan beberapa fungsi tambahan untuk menunjang kinerja dari *Tesseract Library*. Alat *Tesseract* untuk Android

adalah dari seperangkat API Android dan *build file* untuk *Tesseract OCR* dan *Leptonica Library* untuk membantu pengolahan citra [17].

Modul *Tess-two* berisi alat untuk mengompilasi *Tesseract Library* dan *Leptonica* untuk digunakan pada platform Android. Ini menyediakan Java API untuk mengakses API *Tesseract* dan *Leptonica* yang dikompilasi secara native.

### 2.12.2 Leptonica

*Leptonica* merupakan *library* open source yang dapat digunakan sebagai pengolah citra dengan merubah atau mengolah citra ke dalam bentuk *binary image* untuk mengatur tingkat kontras, kecerahan dan perspective citra. *Leptonica* berisi perangkat lunak yang secara luas berguna untuk membangun aplikasi pengolahan citra dan analisis citra [18].

*Leptonica* berasal dari bahasa Italia yang berarti "*leptonic*", yang adalah merupakan kata sifat yang mengacu pada tiga *family* partikel fundamental yang mirip, yang disebut *lepton*, yang lebih kita kenal dengan elektron yang terdiri dari neutrino.

*Leptonica* memiliki jumlah struktur data yang sangat kecil, dan jumlah operasi yang relatif besar. *Leptonica* menggunakan seluruh pendekatan pemrograman berorientasi objek. Struktur datanya menggunakan siklus hidup dimana mereka diciptakan, bertindak, dan dihancurkan. Implementasi yang dilakukan melalui fungsi yang biasanya namanya dimulai dengan nama struktur data primer yang terlibat. Developer mengikuti seperangkat prinsip-prinsip desain untuk membuat kode yang aman, portabel dan transparan untuk digunakan sebaik mungkin (dengan C). Untuk menulis aplikasi sendiri, kita harus melihat ke dalam direktori program, yang ditulis untuk menguji fungsi *library*.

Fungsi yang ada dibagi menjadi dua *library*: satu level tingkat tinggi yang menggunakan serangkaian set struktur data seperti *Pix*, *Box*, dll, dan *library* level rendah yang hanya menggunakan tipe data C intrinsik. Ketika membangun *library*, kita juga membuat sebuah *library* tunggal yang menggabungkan keduanya. Nama file *library* level rendah terurut dibawah nama file *library* level tinggi. Nama fungsi juga mengikuti pola: nama fungsi level rendah yang dihasilkan dari fungsi level tinggi dengan menghapus awalan objek dan menambahkan "*low*".



Sebagai contoh, *pixScaleToGray4()* panggilan *scaleToGray4Low()* untuk menjalankan eksekusi. Ketika struktur data yang dikumpulkan ke dalam sebuah *array*, nama struct ditambahkan huruf "a" diakhir nama filenya. Sebagai contoh, *struct Pixa* berisi *array struct Pix*. Fungsi level atas adalah *pixRead()* dan *pixWrite()*. Encoders spesifik yang didukung oleh fungsi dalam file dinamakan dengan akhiran "*io.c*".

Fungsi-fungsi ini, yang sebagian besar menggunakan aliran untuk I/O, merupakan hubungan antara struktur data citra, *Pix* dan kode level rendah untuk membaca dan menulis data citra. Ada juga fungsi-fungsi pada level tinggi yang khusus. Contohnya, *tiffio.c* memiliki fungsi untuk membaca dan menulis file yang berisi beberapa citra dan menulis file tiff khusus Tag yang tertanam dalam header. Juga, *jpegio.c* memiliki fungsi untuk membaca file jpeg, pilihan untuk mengkonversi RGB ke 8 bpp dengan colormap, mengurangi sebanyak dua tingkat, dan memberi peringatan jika data yang terkompresi rusak. Selain itu, sistem dapat menulis file PostScript dalam berbagai format, baik tingkat 1 (tak terkompresi) dan tingkat 2 (menggunakan DCT alias JPEG) dan kompresi CCITT-G4 dengan pilihan untuk menggambar melalui mask biner. Ini diimplementasikan di *psio.c*, dimana akan ditemukan sejumlah fungsi yang mendukung untuk kedua format kompresi ini. Berikut adalah beberapa operasi yang disediakan oleh *Leptonica*

- a. Rasterop (*bitmap*)
- b. Transformasi affine (*scaling*, translasi, rotasi, geser) pada gambar dengan kedalaman pixel tertentu
- c. Biner dan *grayscale* morfologi
- d. Seedfill dan komponen yang terhubung
- e. Transformasi citra dengan menggabungkan perubahan skala dan kedalaman *pixel*
- f. *Pixelwise masking, blending, enhancement*, dan aritmatika

### 2.12.3 Tesseract Library

*Tesseract* merupakan salah satu library open source yang dapat digunakan pada OCR sebagai *library* bahasa atau *language* agar OCR dapat mengenali karakter . *Tesseract* pada awalnya dikembangkan sebagai perangkat lunak

berpemilik di Hewlett-Packard antara tahun 1985 hingga 1995. Pada saat ini, untuk pertama kalinya, detail arsitektur dan algoritma dipublikasikan.

*Tesseract* dimulai sebagai sebuah proyek penelitian seorang PhD di HP Labs, Bristol, dan mendapatkan momentum sebagai perangkat lunak dan add-on perangkat keras untuk lini HP scanner flatbed. Proyek ini termotivasi oleh fakta bahwa mesin OCR komersial pada saat itu berada di tahap awal, gagal total dan hanya mementingkan kualitas cetak terbaik. Setelah proyek bersama antara HP Labs Bristol, dan Divisi scanner HP di Colorado, *Tesseract* memimpin secara signifikan dalam akurasi yang lebih baik dibandingkan dengan mesin OCR komersial, tetapi *Tesseract* tetap tidak menjadi suatu produk.

Tahap selanjutnya dari perkembangannya adalah kembali ke HP Labs Bristol untuk investigasi OCR sebagai alat kompresi. Pengembangan lebih berkonsentrasi pada peningkatan efisiensi pada tingkat akurasi. Pada akhir proyek ini, pada akhir tahun 1994, pembangunan berhenti sepenuhnya. Mesin OCR dikirim ke UNLV untuk melakukan tes akurasi pada tahun 1995, di mana mesin tersebut membuktikan bahwa kualitas *Tesseract* lebih baik dibandingkan mesin OCR komersial. Setelah sepuluh tahun tanpa perkembangan apapun, Hewlett Packard dan UNLV merilis *Tesseract* sebagai *open source* pada tahun 2005 [19]. Berikut adalah langkah-langkah pemrosesan gambar yang dilakukan oleh *Tesseract* :

a. *Line Finding*

Teknik ini digunakan untuk mendapatkan garis dari sebuah text pada gambar tanpa harus melakukan *rotation* pada gambar tersebut.

b. *Baseline Fitting*

Setelah garis teks ditemukan, *baseline* dipasang lebih tepat menggunakan *quadratic spline*. Ini adalah yang lain pertama untuk sistem OCR, dan diaktifkan *Tesseract* untuk menangani text dengan *baseline* yang tidak rata.

c. *Fixed Pitch Detection and Chopping*

*Tesseract* menguji *baseline* untuk menentukan apakah mereka adalah sudah berada pada *pitch* yang tepat. Dimana jika *tesseract* menemukan

teks pada *pitch* yang tepat, *Tesseract* akan memotong kata ke dalam karakter.

d. *Proportional Word Finding*

*Tesseract* memecahkan sebagian besar masalah ini dengan mengukur kesenjangan dalam jangkauan vertikal terbatas antara garis dasar dan rata-rata garis.

e. *Word Recognition*

Bagian dari adalah proses untuk mengidentifikasi bagaimana seharusnya sebuah kata tersegmentasi menjadi karakter.

f. *Chopping Joined Characters*

Pada tahap ini *Tesseract* meningkatkan hasil pengenalannya dengan cara memotong *blob* yang memiliki hasil pengklasifikasi karakter terburuk. *Blob* adalah wilayah kecil yang terisolasi dari gambar yang dipindai.

g. *Associating Broken Characters*

Tahap ini dilakukan untuk mencari hasil pencari terbaik dari setiap hasil segmentasi dari setiap kandidat karakter

h. *Static Character Classifier*

Pada tahap ini *tesseract* melakukan ekstraksi fitur dan klasifikasi karakter dari teks menggunakan data yang sudah di *training* sebelumnya.

i. *Linguistic Analysis*

Pada tahap ini akan memilih kemungkinan terbaik dari setiap kata yang terdapat pada setiap kategori yang tersedia.

j. *Adaptive Classifier*

Pada tahap ini hasil *ouput* terburuk dari static character classifier akan dilakukan *training* ulang.

#### 2.12.4 TarsosDSP

TarsosDSP adalah *library* Java untuk membantu pemrosesan audio. Tujuannya adalah untuk menyediakan layanan antarmuka yang mudah digunakan untuk menerapkan algoritma pemrosesan musik praktis, sederhana mungkin, dalam Java murni dan tanpa ketergantungan pada bagian eksternal lainnya. *Library* ini mencoba untuk mencapai titik yang cukup mampu untuk menyelesaikan tugas-

tugas nyata tetapi cukup kompak dan sederhana untuk berfungsi sebagai demonstrasi tentang bagaimana algoritma DSP bekerja. TarsosDSP fitur implementasi dari detektor onset perkusi dari sejumlah algoritma deteksi *pitch*: YIN, metode *Mcleod Pitch* dan algoritma " *Dynamic Wavelet Algorithm Pitch Tracking*". Juga termasuk algoritma decoding *Goertzel DTMF*, algoritma *time stretch* (WSOLA), *resampling*, *filter*, sintesis sederhana, beberapa efek audio, dan algoritma *pitch shifting* [18].

### 2.12.5 Android Studio

Android Studio adalah Lingkungan Pengembangan Terpadu - *Integrated Development Environment* (IDE) untuk pengembangan aplikasi Android, berdasarkan *IntelliJ IDEA* [19]. Selain merupakan editor kode dan alat pengembang yang berdaya guna tinggi, Android Studio juga menawarkan fitur lebih banyak untuk meningkatkan produktivitas saat membuat aplikasi Android, misalnya:

- a. Sistem versi berbasis *Gradle* yang fleksibel
- b. *Emulator* yang cepat dan kaya fitur
- c. Lingkungan yang menyatu untuk pengembangan bagi semua perangkat Android
- d. *Instant Run* untuk mendorong perubahan ke aplikasi yang berjalan tanpa membuat *APK* baru
- e. *Template kode* dan integrasi *GitHub* untuk membuat fitur aplikasi yang sama dan mengimpor kode contoh
- f. Alat pengujian dan kerangka kerja yang ekstensif
- g. Alat Lint untuk meningkatkan kinerja, kegunaan, kompatibilitas versi, dan masalah-masalah lain
- h. Dukungan C++ dan NDK
- i. Dukungan bawaan untuk *Google Cloud Platform*, mempermudah pengintegrasian *Google Cloud Messaging* dan *App Engine*

### 2.12.6 Extensible Markup Language (XML)

XML merupakan singkatan dari *Extensible Markup Language*. Pengembangan dan penggunaannya telah mengikuti jalur umum dalam perangkat lunak dan dunia IT. Ini dimulai lebih dari sepuluh tahun yang lalu dan pada awalnya

sangat sedikit yang menggunakannya, tetapi kemudian menarik perhatian publik dan mulai merasuki dunia pertukaran data. Setelah itu, utilitas yang tersedia untuk memproses dan mengelola XML menjadi lebih canggih, sedemikian rupa sehingga banyak orang mulai menggunakannya tanpa benar-benar menyadari keberadaannya. Akhir-akhir ini telah terjadi sedikit reaksi di tempat tertentu atas kegagalan yang dirasakan dan titik lemah, yang telah menyebabkan alternatif yang diusulkan berbagai perbaikan.

Namun demikian, XML sekarang memiliki tempat permanen di sistem TI dan sulit untuk membayangkan aplikasi non-sepele yang tidak menggunakan XML baik untuk konfigurasi atau data sampai tingkat tertentu. Untuk alasan ini sangat penting bahwa pengembang perangkat lunak modern memiliki pemahaman menyeluruh tentang prinsip-prinsipnya, apa yang ia mampu, dan bagaimana menggunakannya untuk keuntungan mereka.

Ada dua penggunaan utama untuk XML, yang pertama adalah cara untuk mewakili *low-level data*, contohnya adalah file konfigurasi. Yang kedua adalah cara untuk menambahkan *metadata* ke dokumen, misalnya seseorang mungkin ingin menekankan kalimat tertentu dalam sebuah laporan dengan menempatkan dalam huruf miring atau tebal. Penggunaan pertama untuk XML dimaksudkan sebagai pengganti cara yang lebih tradisional ini telah dilakukan sebelumnya, biasanya dengan cara daftar pasangan nama / nilai seperti yang terlihat di file INI di Windows atau file Properti Java. Penerapan kedua XML mirip dengan bagaimana HTML file [22].

