

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang Masalah

Hasil dari majunya teknologi informasi yakni penggunaan *smartphone* pada masyarakat luas. Tidak hanya pada kalangan dewasa tapi anak-anak juga sehingga *smartphone* telah menjadi salah satu gaya hidup dan pilihan bantuan untuk beraktivitas tiap harinya. “Fungsi *smartphone* kini juga sudah mengganti beragam gadget seperti jam alarm, konsol game dan digital kamera. Beragam perangkat tersebut kini sudah bisa menyatukan fungsinya di *smartphone*“. Sampai sekarang *smartphone* telah beredar luas dengan berbagai macam jenis sistem operasi. Berdasarkan data yang dikutip dari comScore sebuah situs organisasi layanan analisa data di Amerika memberikan informasi penggunaan perangkat *smartphone* dengan OS Android yang semakin tinggi dari desember 2011 sampai desember 2012. Dikutip juga dari *dailysocial.net* , lembaga riset pasar GfK merilis perhitungannya untuk penjualan *smartphone* di Asia Tenggara sepanjang periode Januari-Maret 2013 (Kuartal Pertama/Q1). Menurut perhitungan GfK, seperti dikutip dari CNET Asia, dari 12,8 juta *smartphone* yang terjual di semua negara Asia Tenggara, 8,8 juta di antaranya (69%) merupakan *smartphone* Android. Dari data tersebut dilansir juga penjualan Android Indonesia mendominasi dengan angka 51% dari total keseluruhan *smartphone*.

Berdasarkan data dari comScore, *dailysocial.net* dan GfK, penggunaan *smartphone* dengan basis Android semakin banyak dan diikuti juga dengan banyaknya aplikasi yang beredar di *playstore*. Buku dan referensi merupakan bagian kategori aplikasi di *playstore*. “Buku merupakan kumpulan kertas atau bahan lainnya yang dijilid menjadi satu pada salah satu ujungnya dan berisi tulisan atau gambar namun seiring perkembangan teknologi maka adapun istilah buku elektronik (*ebook*) yang mengandalkan perangkat komputer, *smartphone*, dan tablet” . Dengan menggunakan aplikasi buku elektronik yang sekarang juga orang pergi ke

gereja hanya membawa *smartphone* yang sudah di lengkapi dengan aplikasi ALKITAB elektrik didalamnya.

Gereja Masehi Injili di Minahasa (GMIM) adalah organisasi gereja yang menerbitkan buku nyanyian NNBT (Nyanyikanlah Nyanyian Baru bagi Tuhan). NNBT adalah kumpulan lagu-lagu abad kini yang di buat oleh warga GMIM yang diteruskan oleh komisi Music Gereja atau Muger GMIM.

Dalam hal ini penulis akan merancang aplikasi buku elektronik nyanyian gerejawi yang berisi lagu-lagu dalam buku nyanyian NNBT yang berbasis android. Aplikasi ini akan mempermudah jemaat untuk mempelajari lirik lagu maupun mempelajari nada dari lagu-lagu NNBT. Dengan ini penulis memberi judul dalam pembuatan tugas akhir adalah “**Aplikasi Buku Elektronik Nyanyian NNBT berbasis android**”.

## **1.2. Rumusan Masalah**

Berdasarkan latar belakang yang telah di uraikan di atas maka permasalahan yang dapat di identifikasi adalah

“Bagaimana merancang aplikasi sebuah aplikasi Buku nyanyian NNBT untuk jemaat GMIM di *smartphone* berbasis android. ? ”

## **1.3. Batasan Masalah**

Permasalahan dalam penelitian ini dibatasi pada pembuatan aplikasi buku nyanyian NNBT berbasis android, dengan ketentuan sebagai berikut :

1. Nyanyian yang disajikan hanya berbentuk lirik , not balok, midi.
2. Multimedia yang di gunakan yaitu : text, citra dan audio.

## **1.4. Tujuan Penelitian**

Secara umum, penelitian dan penulisan ini dibuat guna memenuhi salah satu syarat untuk mendapatkan gelar kesarjanaan pendidikan D4 di Politeknik Negeri Manado.

Dan secara khusus, penelitian ini bertujuan untuk :

- Merampungkan satu buah buku NNBT berjumlah 50 buah lagu dalam satu aplikasi Mobile SmartPhone Android.

- Jemaat dapat belajar menyanyikan lagu NNBT dengan tepat dan benar secara individual.
- Dan di tujukan agar dapat membantu jemaat saat ibadah.

### **1.5. Manfaat Penelitian**

Manfaat yang diharapkan dari pembuatan media ini adalah :

- a. Peneliti mendapat ilmu pengetahuan tentang cara membuat aplikasi buku nyanyian NNBT berbasis android ini.
- b. Peneliti juga bisa mendapatkan ilmu pengetahuan tentang music terlebih khusus dalam membaca nada dan pembuatan midi lagu.

### **1.6. Sistematika Penulisan**

Sistematika Penulisan ini terdiri dari Lima Bab yang terdiri dari:

1. BAB 1: pada bab berisi tentang pendahuluan, permasalahan yang dihadapi, tujuan dari penelitian, serta ruang lingkup dan penelitian. Dan pada bagian akhir bab ini diuraikan sistematika penyajian laporan penelitian ini.
2. BAB 2: pada bagian ini berisi tentang landasan teori dalam pembuatan tugas akhir yang membahas tentang Buku NNBT, Android, Notasi Balok, Eclipse IDE, *android developer tools*, Finale Music Maker, Midi, Partitur, Java dan *Smart Phone*.
3. BAB 3: pada bagian ini berisi tentang metode penelitian, analisis masalah, metode pengujian.
4. BAB 4: bagian ini berisi tentang kesimpulan dari hasil penelitian penulis dan saran untuk pembuatan lebih lanjut.
5. BAB 5: bagian ini berisi tentang Kesimpulan dan saran tugas akhir yang dibuat.

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1. Buku Nyayikanlah Nyanyian Baru bagi Tuhan (NNBT)**

Dengan melalui Komisi Pengembangan Kesenian Sinode GMIM, melaksanakan satu kegiatan Lokakarya dan Pelatihan Mencipta Lagu yang dilaksanakan pada tanggal 22 – 26 Oktober 2002 di jemaat “EBEN HEAZER” Woloan Wilayah Tomohon II, berkerja sama dengan Yayasan SERAFICA Jakarta. Sebagian lagu ini adalah hasil dari lokakarya tersebut ditambah dengan beberapa lagu yang telah terkumpul sejak tahun 1999. Dan kemudian pada tahun 2010 buku nyanyian. Setelah peluncuran pada tahun 2010, telah di dapati beberapa kesalahan radikal yang harus di perbaiki. Namun proses koreksi sebagaimana dimaksud pada NNBT Cetakan Kedua nanti dilakukan pada tahun 2011. Pada tahun yang kedua NNBT melakukan pencetakan ketiga yang hadir dengan tampilan yang berbeda, baik dari segi ukurannya maupun isinya.

#### **2.2. Android**

Android adalah [sistem operasi](#) berbasis [Linux](#) yang dirancang untuk perangkat bergerak [layar sentuh](#) seperti [telepon pintar](#) dan [komputer tablet](#). Android awalnya dikembangkan oleh Android, Inc., dengan dukungan finansial dari [Google](#), yang kemudian membelinya pada tahun 2005. Sistem operasi ini dirilis secara resmi pada tahun 2007, bersamaan dengan didirikannya [Open Handset Alliance](#), konsorsium dari perusahaan-perusahaan [perangkat keras](#), perangkat lunak, dan telekomunikasi yang bertujuan untuk memajukan [standar terbuka](#) perangkat seluler. [Ponsel Android pertama](#) mulai dijual pada bulan Oktober 2008.

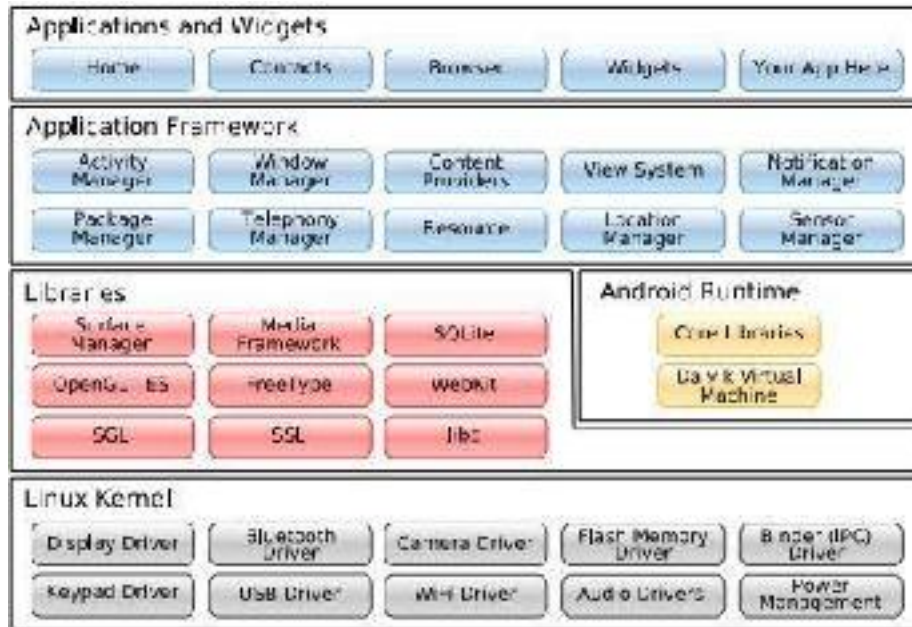
[Antarmuka pengguna](#) Android umumnya berupa [manipulasi langsung](#), menggunakan gerakan sentuh yang serupa dengan tindakan nyata, misalnya menggeser, mengetuk, dan mencubit untuk memanipulasi objek di layar, serta [papan ketik virtual](#) untuk menulis teks. Selain perangkat layar sentuh, Google juga telah mengembangkan [Android TV](#) untuk televisi, [Android Auto](#) untuk mobil, dan [Android Wear](#) untuk jam tangan, masing-masingnya memiliki antarmuka pengguna yang berbeda. Varian Android juga digunakan pada [komputer jinjing](#), [konsol permainan](#), [kamera digital](#), dan peralatan elektronik lainnya.

Android adalah sistem operasi dengan [sumber terbuka](#), dan Google merilis kodenya di bawah [Lisensi Apache](#). Kode dengan sumber terbuka dan lisensi perizinan pada Android memungkinkan perangkat lunak untuk dimodifikasi secara bebas dan didistribusikan oleh para pembuat perangkat, operator nirkabel, dan pengembang aplikasi. Selain itu, Android memiliki sejumlah besar komunitas pengembang aplikasi ([apps](#)) yang memperluas fungsionalitas perangkat, umumnya ditulis dalam versi kustomisasi bahasa pemrograman [Java](#). Pada bulan Oktober 2013, ada lebih dari satu juta aplikasi yang tersedia untuk Android, dan sekitar 50 miliar aplikasi telah diunduh dari [Google Play](#), toko aplikasi utama Android. Sebuah survei pada bulan April-Mei 2013 menemukan bahwa Android adalah platform paling populer bagi para pengembang, digunakan oleh 71% pengembang aplikasi bergerak. Di [Google I/O 2014](#), Google melaporkan terdapat lebih dari satu miliar pengguna aktif bulanan Android, meningkat dari 583 juta pada bulan Juni 2013.

Faktor-faktor di atas telah memberikan kontribusi terhadap perkembangan Android, menjadikannya sebagai sistem operasi telepon pintar yang paling banyak digunakan di dunia, mengalahkan [Symbian](#) pada tahun 2010. Android juga menjadi pilihan bagi perusahaan teknologi yang menginginkan sistem operasi berbiaya rendah, bisa dikustomisasi, dan ringan untuk perangkat berteknologi tinggi tanpa harus mengembangkannya dari awal. Sifat Android yang terbuka juga telah mendorong munculnya sejumlah besar komunitas pengembang aplikasi untuk menggunakan kode sumber terbuka sebagai dasar proyek pembuatan aplikasi, dengan menambahkan fitur-fitur baru bagi pengguna tingkat lanjut atau mengoperasikan Android pada perangkat yang secara resmi dirilis dengan menggunakan sistem operasi lain.

Pada November 2013, Android menguasai pangsa pasar telepon pintar global, yang dipimpin oleh produk-produk [Samsung](#), dengan persentase 64% pada bulan Maret 2013. Pada Juli 2013, terdapat 11.868 perangkat Android berbeda dengan beragam versi. Keberhasilan sistem operasi ini juga menjadikannya sebagai target litigasi paten "[perang telepon pintar](#)" antar perusahaan-perusahaan teknologi. Hingga bulan Mei 2013, total 900 juta perangkat Android telah diaktifkan di seluruh dunia, dan 48 miliar aplikasi telah dipasang dari Google Play.

Sistem operasi Android dibagi menjadi 5 bagian dalam 4 lapisan utama.



Gambar 2.1 Arsitektur Android

Berikut penjelasan dari Gambar 2.3. Linux *kernel*, lapisan ini berisi semua perangkat *driver lowlevel* untuk berbagai macam komponen *hardware* dari perangkat Android.

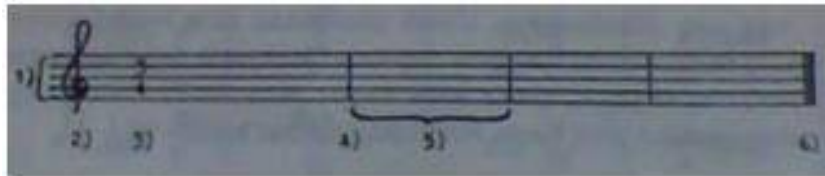
1. *Libraries*, lapisan ini berisi semua *code* yang menyediakan fitur-fitur utama dari sistem operasi Android. Contohnya *library SQLite* menyediakan dukungan *database* sehingga aplikasi bisa menggunakannya untuk penyimpanan data.
2. *Android runtime*, lapisan ini berada pada lapisan yang sama dengan *libraries*. Lapisan ini menyediakan satu set *library* yang memungkinkan *developer* untuk membuat aplikasi Android menggunakan bahasa pemrograman Java. Lapisan ini juga mencakup mesin virtual Dalvik, yang memungkinkan setiap aplikasi Android untuk berjalan dalam prosesnya sendiri dengan *instance*-nya masing-masing dalam mesin virtual Dalvik. Dalvik adalah mesin virtual khusus yang dirancang terutama untuk Android dan dioptimalkan untuk perangkat *mobile* bertenaga baterai dengan memori dan CPU yang terbatas.
3. *Application framework*, lapisan ini memaparkan berbagai kemampuan sistem operasi Android kepada *developer* aplikasi sehingga *developer* bisa menggunakannya pada aplikasi yang dikembangkan.

4. *Applications*, pada lapisan teratas ini, terdapat aplikasi yang sudah ada pada perangkat Android, seperti telepon, kontak, browser, dan aplikasi lain yang langsung dapat digunakan oleh pengguna. Pada lapisan ini juga terdapat aplikasi-aplikasi yang di-*download* dan di-*install* dari Android Market. Setiap aplikasi yang dibuat ditempatkan pada lapisan ini.

### 2.3. Notasi Balok

Notasi balok adalah Sistem penulisan lagu atau karya musik lainnya yang dituangkan dalam bentuk gambar. Gambar-gambar yang melambangkan bunyi tersebut dituliskan dalam Not balok sesuai dengan tinggi-rendah dan sifat bunyi yang dilambangkan.

Dalam materi not balok kita mengenal beberapa istilah yang terdapat didalam materii-materi not balok, seperti paranada, tanda kunci, birama, garis birama, bar, dan garis penutup. Perhatikan gambar berikut.



Gambar 2.2 Contoh notasi balok

Keterangan gambar :

1. Paranada adalah lima garis lurus yang berjajar mendatar dan berjarak sama. Paranada digunakan untuk menuliskan lambang-lambang bunyi sesuai dengan sifat nada yang dilambangkan.
2. Tanda kunci adalah tanda untuk menetapkan letak salah satu [nada](#) dalam Not-balok. Untuk menulis partitur [gitar](#) digunakan tanda kunci G, artinya dalam not balok tersebut nada G terletak pada garis ke-2.
3. Birama adalah gerak melody yang teratur dalam sebuah lagu atau karya musik lainnya. dalam contoh diatas dituliskan birama 3/4 artinya dalam setiap ruas birama bernilai tiga ketuk dengan satuannya adalah not 1/4.

4. Garis birama adalah garis yang dituliskan secara tegak lurus dengan paranada yang berfungsi untuk membatasi antar ruas birama yang satu dengan ruas birama yang lainnya.
5. Bar (ruas birama) adalah ruas yang terdapat diantara dua garis birama. bar berfungsi untuk menuliskan not sesuai dengan birama yang dipergunakan. nilai not dalam suatu bar selalu sama. mungkin sobat sering menjumpai nilai not yang terdapat pada bar pertama dan bar terakhir berbeda dengan bar-bar lainnya. tetapi apabila nilai not yang terdapat pada kedua bar tersebut sobat jumlahkan, maka akan sama dengan bar-bar lainnya.
6. Garis penutup adalah dua buah garis tebal tipis yang dituliskan tegak lurus dengan paranada yang berfungsi sebagai petunjuk berakhirnya sebuah lagu atau karya musik lainnya.













Sebelum kita belajar jauh mengenai notasi balok maka terlebih dahulu kita harus benar-benar memahami nama dan letak not dalam paranada. Letak not dan paranada sangat tergantung pada dari sifat nada yang dilambangkan. untuk menuliskan nada rendah ataupun nada tinggi, dimana 5 garis paranada sudah tidak dapat lagi mampu menampungnya, maka kita dapat menuliskannya dengan menggunakan garis-garis bantu not dalam paranada, Perhatikan gambar berikut ini.



Gambar 2.3 Garis bantu not dalam paranada

Sebuah not dapat mempunyai nilai  $\frac{1}{2}$  ketuk, 1 ketuk, 2 ketuk, 4 ketuk dan sebagainya. Hal ini tergantung dari birama yang dipergunakan sebagai satuan ketuknya. Dalam bab ini kita akan mempelajari bentuk dan nilai not serta tanda diam dengan menggunakan birama per-empat. artinya, Not yang dijadikan sebagai satuan ketuknya adalah Not  $\frac{1}{4}$ . Perhatikan gambar berikut.



Nama Not	Bentuk Not	Tanda Istirahat	Nilai
Not Penuh			4 Ketuk
Not 1/2			2 Ketuk
Not 1/4			1 Ketuk
Not 1/8			1/2 Ketuk
Not 1/16			1/4 Ketuk
Not 1/32			1/8 Ketuk

Gambar 2.4 Bentuk dan nilai not serta tanda diam

## 2.4. Eclipse IDE

Eclipse adalah sebuah *open source software* yang gunanya menyediakan *tool platform* yang sangat terintegrasi [9]. Dalam penggunaannya Eclipse meliputi *core project* dimana yang berisi *generic framework* untuk *tool integration* dan sebuah *Java development environment*. Proyek pada Eclipse terimplementasikan pada Java dan dapat dijalankan pada beberapa sistem operasi meliputi Windows, Mac OSX dan Linux.

Proyek Eclipse mendukung pengembangan dari sebuah *platform* atau *framework* untuk pengimplementasian dari *Integrated Development Environment (IDE)* dan aplikasi lainnya. Meskipun *framework* dari Eclipse diimplementasikan pada Java tidak menutup kemungkinan pengembangan juga dilakukan pada bahasa pemrograman lainnya seperti C++, XML dan sebagainya.

Eclipse adalah sebuah [IDE](#) (*Integrated Development Environment*) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua platform (*platform-*

*independent*). Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti C/C++, [Cobol](#), [Python](#), [Perl](#), [PHP](#), dan lain sebagainya. Selain sebagai IDE untuk pengembangan aplikasi, Eclipse pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, test perangkat lunak, pengembangan web, dan lain sebagainya. Eclipse pada saat ini merupakan salah satu IDE favorit dikarenakan **gratis dan open source**, yang berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu, kelebihan dari Eclipse yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan *plug-in*.

## 2.5. Android Developer Tools

Android Development Tools (ADT) adalah plugin yang didesain untuk IDE Eclipse yang memberikan kita kemudahan dalam mengembangkan aplikasi android dengan menggunakan IDE Eclipse. Dengan menggunakan ADT untuk Eclipse akan memudahkan kita dalam membuat aplikasi project android, membuat GUI aplikasi, dan menambahkan komponen-komponen yang lainnya, begitu juga kita dapat melakukan running aplikasi menggunakan Android SDK melalui Eclipse. Dengan ADT juga kita dapat melakukan pembuatan package android (.apk) yang digunakan untuk distribusi aplikasi android yang kita rancang.

Semakin tinggi platform android yang kita gunakan, dianjurkan menggunakan ADT yang lebih terbaru, karena biasanya munculnya platform baru diikuti oleh munculnya versi ADT yang terbaru. Untuk melakukan instalasi ADT di-eclipse dapat dilakukan secara on-line maupun offline

## 2.6. Finale Music Maker

**Finale** adalah program unggulan dari serangkaian proprietary scorewriter music notasi perangkat lunak yang dibuat oleh MakeMusic untuk [Microsoft Windows](#) dan [OS X](#). Pertama kali dirilis pada tahun 1988, versi 2014 dirilis pada 2013. Finale dianggap sebagai salah satu standar industri untuk perangkat lunak notasi musik.

Finale adalah salah satu dari sejumlah jenis [software](#) yang digunakan oleh [komposer](#), [penulis lagu](#) dan [arranger](#) untuk menciptakan [lembaran musik](#), termasuk skor untuk seluruh ensemble (misalnya, [orkestra](#), [band konser](#), [big band](#), dll) dan bagian untuk musisi individu. Sebuah scorewriter adalah [notasi musik](#) apa [pengolah kata](#) adalah teks, di bahwa mereka berdua memungkinkan koreksi cepat (melalui "[membatalkan](#)" tombol), editing fleksibel, mudah

berbagi konten (melalui internet atau compact [media penyimpanan](#)), dan produksi bersih, tata letak seragam. Selain itu, sebagian scorewriters, termasuk Finale, dapat menggunakan perangkat lunak berbasis [synthesizer](#) untuk "bermain" suara musik dinotasikan dan merekam musik-an fitur terutama berguna untuk pemula [komposer](#) , ketika tidak ada [musisi](#) yang tersedia, atau jika komposer tidak mampu untuk menyewa musisi.

Alat Finale ini akan disusun dalam beberapa palet hirarki terorganisir, dan alat yang sesuai harus dipilih untuk menambah atau mengubah setiap kelas tertentu dari unsur skor, (misalnya, alat Shape Cerdas untuk menghasilkan dan mengedit [getar](#) garis dan [dinamika](#) "jepit rambut" (dinamakan demikian karena simbol-simbol menyerupai [pin rambut](#) ); alat Staf untuk menambah dan mengedit parameter tongkat individu). Di samping alat-alat ini, kontrol tambahan yang tersedia untuk melihat atau menyembunyikan hingga empat lapisan ditumpangkan musik yang dapat dimasukkan ke setiap staf tertentu, untuk tujuan pengorganisasian beberapa suara kontrapungtal pada staf yang sama. Beberapa alat Finale ini menyediakan menu yang terkait hanya ke kiri dari menu Help, tersedia hanya ketika bahwa alat tertentu dipilih. Dengan demikian, operasi Finale dikenakan setidaknya beberapa kesamaan permukaan untuk [Adobe Photoshop](#) .

Pada layar, Finale menyediakan kemampuan untuk kode warna beberapa elemen dari skor sebagai bantuan visual; di print-out semua mencetak elemen hitam (kecuali warna print-out secara eksplisit dipilih). Dengan alat yang sesuai dipilih, penyesuaian halus dari setiap set objek dalam skor yang mungkin baik dengan mengklik dan menyeret atau dengan memasukkan pengukuran dalam kotak dialog. Sebuah alat seleksi yang lebih umum juga tersedia untuk memilih daerah ukuran besar untuk kunci editing dan tanda tangan waktu, atau transposing, antara penggunaan lainnya. Alat ini juga menyediakan kemampuan untuk mereposisi beberapa kelas objek skor langsung, dan versi yang lebih baru dari perangkat lunak telah dilaksanakan menuing kontekstual luas melalui alat ini.

Finale otomatis mengelola banyak aturan dasar harmoni dan musik notasi, seperti arah yang benar batang, alignment vertikal beberapa nilai berirama, dan aturan yang ditetapkan untuk posisi noteheads chords. Dalam situasi lain, tanpa muka pengguna kustomisasi hati, program ini membuat apa yang dapat digambarkan sebagai *tebakan yang baik*, terutama di bidang [ejaan ENHARMONIC](#) data baru masuk dihasilkan dari [MIDI](#) keyboard, sementara menghormati tanda kunci saat ini. Ini adalah "pintar" cukup untuk mengeja sebuah [ENHARMONIC](#) lapangan

ketika [dominan sekunder](#) yang digunakan dalam sepotong. Di kunci c minor, jika musisi memainkan akord dominan dari chord dominan, D dominan 7 (D dominan 7 adalah dominan G utama, yang merupakan dominan c minor), Finale akan mengeja D 7 chord dengan F-tajam; tapi untuk akord lainnya, seperti G utama 7 5 (chord ketujuh utama dengan kelima ditambah), mungkin kadang-kadang menggunakan G-datar bukannya F-tajam.

Bagi sebagian besar musik tonal Barat, Finale memilih ejaan yang benar untuk akord dari tonik dan kunci dominan dengan benar, tetapi ketika musik mengembara ke daerah tonal lebih jauh dari tonik, Finale cenderung membuat ENHARMONIC "ejaan" kesalahan dengan memperlakukan chords sebagai jika mereka milik kunci tonik dalam beberapa cara. Bila menggunakan kunci yang tidak standar, para ahli telah merekomendasikan bahwa pengguna "menetapkan ejaan untuk setiap lapangan dalam skala kromatik" menggunakan kotak dialog yang tersedia dari menu Preferences.

## 2.7. MIDI

MIDI (*Musical Instrument Digital Interface*) adalah bahasa yang digunakan instrumen musik elektrik, [pengendali](#), komputer, dan peranti sejenis untuk berkomunikasi. MIDI juga mengandung spesifikasi peranti keras yang memungkinkan alat-alat tersebut terhubung. MIDI menangkap event notasi dan perubahan atribut dan aksentuasi nada, mengkodekannya menjadi pesan digital, dan mengirimkan kode tersebut sebagai pesan ke peranti lain untuk mengatur suara yang dihasilkan beserta parameternya. Data jenis ini dapat direkam dengan [sequencer](#), seperti Cubase, Fruity Loops, atau Finale. MIDI membawa pesan event musikal yang terdiri dari [notasi](#), [pitch](#) and velocity (tekanan), sinyal pengendali seperti [volume](#), [vibrato](#), [audio panning](#) dan [cues](#), juga [clock signals](#) yang diatur mensinkronkan [tempo](#) antara berbagai peranti musik. Satu file MIDI dapat memuat enam belas channel informasi musik yang masing-masing dapat di arahkan ke peranti yang berbeda. Dari enam belas channel tersebut, channel sepuluh khusus dipergunakan untuk perkusi. MIDI berperan besar pada revolusi industri rekaman pada saat diperkenalkan pada tahun 1983.

## 2.8. PARTITUR

Partitur (Partiture, dalam bahasa Inggris) merupakan suatu istilah umum yang mendeskripsikan atau merujuk mengenai kertas musik atau standar music notation. Ada juga istilah lain yang kita kenal seperti Score, Sheet, lembran not balok, dan lain sebagainya. Merujuk

kepada satu maksud yaitu notasi musik (music notation). Tujuan pembuatan lembaran musik dapat digunakan sebagai catatan dari, panduan, atau sarana untuk melakukan, sepotong musik.

## **2.9. JAVA**

Bahasa Java diciptakan pertama kali oleh sekelompok tim yang dipimpin oleh James Gosling di Sun Microsystems yang pertama kalinya disebut sebagai Oak yang diciptakan tahun 1991 untuk digunakan pada *chip* yang ditanamkan di peralatan elektronik. Java itu sederhana, *object oriented*, aman, *portable*, memiliki performansi yang tinggi, *multithreaded*, dan dinamis. Java juga merupakan bahasa pemrograman yang penuh fitur dan dapat digunakan untuk membuat berbagai aplikasi baik yang umum maupun yang tingkat tinggi. Sekarang ini java digunakan tidak hanya untuk pemrograman web tetapi juga untuk membuat aplikasi yang bisa berdiri sendiri meskipun berbeda perangkat seperti pada server, komputer, ataupun pada perangkat mobile.

## **2.10. SMARTPHONE**

*Smartphone* menyediakan fasilitas internet dan akses ke layanan data yang beragam. Teknologi *smartphone* telah pindah ke perangkat yang sangat visual dan menarik banyak pengguna, serta memberikan banyak fasilitas untuk berkomunikasi. Beberapa sistem operasi yang sedang berkembang saat ini adalah Blackberry, iOS, dan Android.

## **2.11. Unified Modeling Language (UML)**

Untuk memudahkan komunikasi antar pengembang perangkat lunak diperlukan adanya satu spesifikasi standar. Para pengembang perangkat lunak harus mengikuti standar dan petunjuk yang telah ditetapkan pada proyek pengembangan perangkat lunak, dan beberapa petunjuk tersebut direpresentasikan dalam model.

Model direpresentasikan dengan notasi umum dan model yang standar akan lebih mudah dipahami dan digunakan. Model ini akan meningkatkan komunikasi antar pengembang dan mengurangi kesalahan terjadinya kesalahpahaman saat mengembangkan suatu perangkat lunak. Menggunakan model ini juga akan menghemat waktu karena model telah memberikan jawaban atas pilihan, dan menyebabkan pengembang lebih fokus terhadap pengembangan. UML adalah salah satu model yang standar dan banyak digunakan oleh pengembang perangkat lunak. UML

mendefinisikan notasi dengan sangat baik dan memiliki arti yang sangat mendekati pemodelan berorientasi objek.

### 2.11.1 Use-Case Diagram

*Use-Case Diagram* menampilkan relasi antara aktor dengan *Use-Case* dalam suatu sistem. *Use-Case Diagram* digunakan untuk:

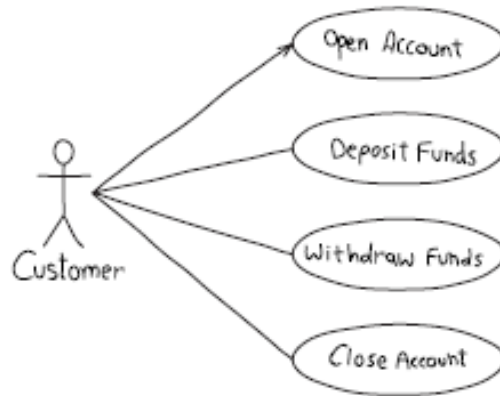
1. Memberikan Gambaran keseluruhan dari kebutuhan sistem atau organisasi dalam bentuk model,
2. Mengkomunikasikan ruang lingkup dari suatu proyek pengembangan,
3. Memodelkan analisa kebutuhan di dalam suatu *Use-Case* model

*Use-Case Diagram* sebaiknya dibuat berdasarkan sudut pandang dari *stakeholder*.

Elemen-elemen yang terdapat pada *Use-Case diagram* adalah sebagai berikut:

1. *Use-Case*, mendeskripsikan sekumpulan aksi yang dilakukan oleh aktor dan digambarkan dalam bentuk *eclipse*.
2. *Actor*, adalah satu orang, organisasi, proses lokal atau sistem eksternal yang memainkan suatu peran dalam satu atau lebih interaksi dengan sistem dan digambarkan dengan bentuk figure orang.
3. *Relationship*, relasi antara *Actor* dengan *Use-Case* yang digambarkan dengan garis lurus.
4. *System Boundary*, kotak disekeliling *Use-Case* yang menandakan ruang lingkup dari sistem. *Use-Case* yang berada dalam *system boundary* merepresentasikan fungsi-fungsi yang akan diimplementasikan.

Pada Gambar 2. 5 Merupakan contoh dari sebuah use-case diagram



Gambar 2.5 Contoh *Use-Case* diagram

### 2.11.2 Class Diagram

*Class Diagram* digunakan untuk menampilkan kelas dari sistem, hubungannya, dan juga operasi serta atribut yang dimiliki oleh kelas tersebut.

*Class Diagram* digunakan untuk:

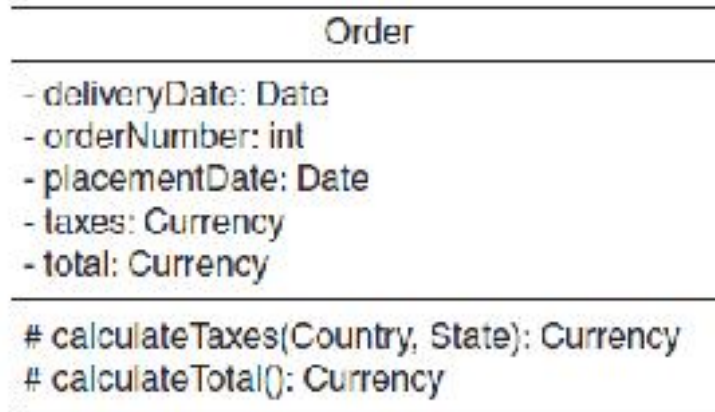
1. Memodelkan suatu konsep,
2. Menganalisa kebutuhan dalam model konseptual,
3. Menggambarkan rancangan sistem atau perangkat lunak berorientasi objek.

Sebuah model terdiri dari satu atau lebih *Class Diagram* dan spesifikasi pendukung yang menjelaskan elemen dari diagram termasuk kelas, relasi antar kelas dan juga antarmuka.

Berikut adalah *visibility options* pada *Class Diagrams*:

1. *Public*, ditandai dengan simbol (+) yang digunakan untuk menandakan bahwa atribut atau operasi dapat diakses oleh seluruh kelas dalam satu sistem.
2. *Protected*, ditandai dengan simbol (#) yang digunakan untuk menandakan bahwa atribut atau operasi hanya dapat diakses oleh kelas itu sendiri dan *subclass*-nya.
3. *Private*, ditandai dengan simbol (-) yang digunakan untuk menandakan bahwa atribut atau operasi hanya dapat diakses oleh kelas itu sendiri.
4. *Default*, ditandai dengan simbol (~) yang digunakan untuk menandakan bahwa atribut atau operasi hanya dapat diakses oleh kelas itu sendiri dan kelas yang terdapat pada *package* yang sama.

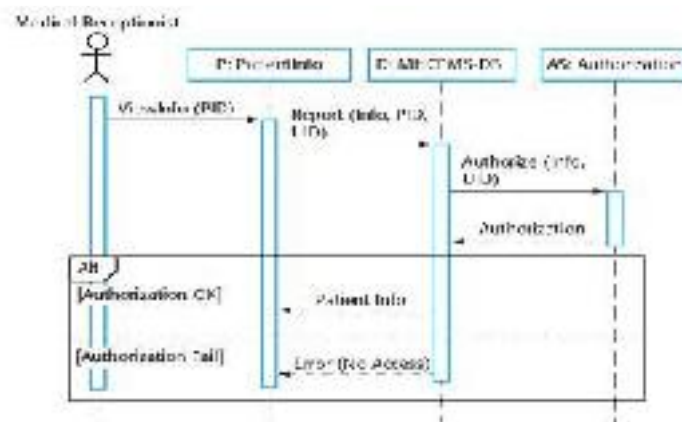
Pada gambar 2.6 merupakan contoh sebuah *class diagram*



Gambaran 2.6 Contoh Class diagram

### 2.11.3 Sequence Diagram

*Sequence Diagram* adalah salah satu teknik pemodelan yang dinamis. *Sequence Diagram* digunakan untuk merepresentasikan interaksi antara komponen sistem dan terkadang agen eksternal juga termasuk di dalamnya. Pada Gambar 2.6 merupakan contoh *sequence diagram*



Gambar 2.7 Contoh Sequence diagram

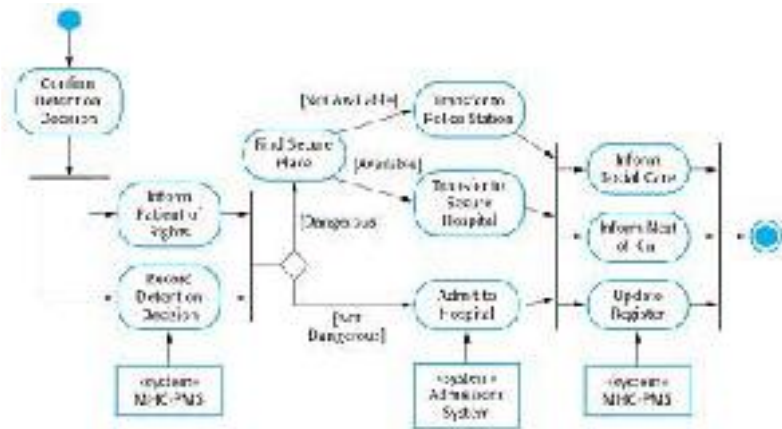
### 2.11.4 Activity Diagram

*Activity Diagram* adalah diagram yang digunakan untuk merepresentasikan:

1. Sebuah operasi yang rumit.
2. Sebuah proses bisnis yang rumit.
3. Sebuah *Use-Case*.
4. Sebuah proses perangkat lunak.



Pada Gambar 2.8 merupakan contoh *activity* diagram.



Gambar 2.8 Contoh *Activity* diagram