

**PERANCANGAN PERANGKAT LUNAK KONVERSI NOTASI ANGKA
KE DALAM NOTASI BALOK DENGAN *VISUAL BASIC***

Tugas Akhir Diajukan untuk Memenuhi Salah Satu Syarat Kelulusan
Program Pendidikan Sarjana

Oleh:

Indra

2011130004



JURUSAN TEKNIK INFORMATIKA
SEKOLAH TINGGI MANAJEMEN INFORMATIKA & KOMPUTER - LIKMI
BANDUNG
2016

PERANCANGAN PERANGKAT LUNAK KONVERSI NOTASI MUSIK
DENGAN *VISUAL BASIC*

Oleh : Indra 2011130004

Bandung, 22 Februari 2016

Menyetujui,

Deddy Koswara, S.KOM., M.T.

Pembimbing

Dhanny Setiawan, S.T., M.T.

Ketua Jurusan

JURUSAN TEKNIK INFORMATIKA SEKOLAH TINGGI MANAJEMEN
INFORMATIKA & KOMPUTER – LIKMI BANDUNG

2016

ABSTRAK

Perkembangan teknologi informasi memberikan manfaat tersendiri pada berbagai aspek kehidupan sehari-hari. Hadirnya teknologi perangkat lunak dibidang musik yang berkembang pesat, membawa dampak positif yang memudahkan seorang musisi maupun penikmat musik dalam membuat maupun mendengarkan musik. Saat ini perkembangan perangkat lunak dari perangkat lunak pengolah suara berkembang sampai perangkat lunak untuk pembuatan musik, pembelajaran *chord* alat musik, tangga nada, sampai pembuatan partitur.

Dengan dasar tersebut, maka dibangunlah sebuah perangkat lunak yang berfungsi sebagai media pembuatan partitur notasi angka maupun notasi balok. Perangkat lunak yang dibangun ini terdapat fitur untuk membuat partitur notasi angka yang dapat dikonversi menjadi partitur notasi balok, serta fitur untuk memainkan nada-nada yang telah dibuat oleh pengguna.

Pembuatan perangkat lunak ini menggunakan metode pengembangan *prototype* dan metodologi pemrograman terstruktur dengan *development tools* yang digunakan adalah Visual Studio 2012 dengan komponen Devexpress.

KATA PENGANTAR

Terima kasih pertama-tama diucapkan kepada Tuhan Yesus Kristus, karena atas segala berkat, hikmah, bimbingan, dan anugerah-Nya maka penulis dapat menyelesaikan penulisan dan perancangan produk Tugas Akhir tepat pada waktunya.

Selama penulisan Tugas Akhir ini, penulis telah mendapatkan banyak saran, bimbingan, kritik, pembelajaran, dan dukungan dari berbagai pihak, sehingga penulis dapat menyelesaikan Tugas Akhir ini. Untuk itu penulis juga mengucapkan banyak terima kasih kepada

1. Bapa Dedi Koswara, S.KOM., M.T. selaku dosen pembimbing yang telah membimbing, memberi saran, mengoreksi, dan meluangkan waktunya untuk membantu penulis menyelesaikan Tugas Akhir ini.
2. Bapak Dhanny Setiawan, S.T., M.T. sebagai ketua jurusan Teknik Informatika yang telah banyak memberi ilmu kepada penulis selama menempuh kuliah di STMIK LIKMI.
3. Keluarga penulis yang selalu mendukung dan memberi semangat selama penulisan Tugas Akhir ini.
4. Seluruh teman-teman jurusan Teknik Informatika angkatan 2011 yang telah memberi semangat kepada penulis.
5. Teman-teman komsel Yehezkiel yang selalu mendukung dan memberi semangat kepada penulis.
6. Seluruh teman-teman gereja GBI PPL Majesty yang selalu mendukung dan memberi motivasi kepada penulis.
7. Seluruh dosen STMIK LIKMI yang selama ini telah mengajarkan banyak ilmu kepada penulis selama masa perkuliahan.
8. Semua pihak yang tidak dapat penulis sebutkan satu per satu.

Akhir kata, penulis memohon maaf jika masih terdapat beberapa kekurangan dalam penulisan Tugas Akhir ini dan semoga dapat bermanfaat bagi pembacanya.

Bandung, 22 Februari 2015

Penulis

DAFTAR ISI

ABSTRAK	i
KATA PENGANTAR	ii
DAFTAR ISI	iv
DAFTAR GAMBAR	vi
DAFTAR TABEL	viii
DAFTAR SIMBOL	ix
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah	2
1.3 Tujuan Penulisan	2
1.4 Batasan Masalah	3
1.5 Kegunaan Hasil	3
1.6 Metodologi Penelitian	3
1.7 Sistematika Penulisan	4
BAB 2 LANDASAN TEORI	5
2.1 Rekayasa Perangkat Lunak	5
2.2 Model <i>Prototype</i>	7
2.3 Pemrograman Terstruktur	9
2.4 Diagram	11
2.4.1 Data Flow Diagram	12
2.4.2 Spesifikasi Proses	14
2.5 Notasi Musik	15
2.6 Elemen Musik	16
2.6.1 Lambang Notasi Musik	16
2.6.2 Tanda Diam	17
2.6.3 Tanda Kromatis	17
2.7 Nilai Nada	19

2.8 Birama dan Irama	21
2.9 Sangkar Nada atau Paranada	23
2.10 Tanda Kunci	24
2.11 Perubahan Nada Dasar	25
2.12 Visul Studio 2012	28
2.13 Text File	30
2.14 DevExpress	31
BAB 3 ANALISIS DAN PERANCANGAN PERANGKAT LUNAK	32
3.1 Gambaran Umum Perangkat Lunak	32
3.2 Spesifikasi Kebutuhan	33
3.2.1 Analisis Fungsional	33
3.2.2 Analisis Non Fungsional	33
3.3 Data Flow Diagram	34
3.4 Spesifikasi Proses	38
3.5 Struktur Data	41
3.6 Tampilan Antar Muka	43
BAB 4 IMPLEMENTASI DAN PENGUJIAN PERANGKAT LUNAK	47
4.1 Spesifikasi Kebutuhan perangkat Lunak	47
4.2 Tampilan Antar Muka	47
4.3 Pengujian Fungsi	54
BAB 5 KESIMPULAN DAN SARAN	59
5.1 Kesimpulan	59
5.2 Saran	59
DAFTAR PUSTAKA	60
LAMPIRAN LISTING PROGRAM	61

DAFTAR GAMBAR


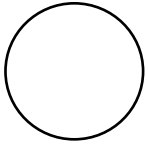
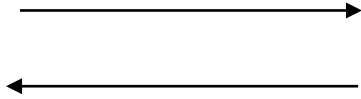
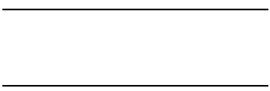
Gambar 2.1 Software Engineering Layer	6
Gambar 2.2 Model Prototype	8
Gambar 2.3 Notasi Balok	17
Gambar 2.4 Tanda Diam	17
Gambar 2.5 Notasi Kres	18
Gambar 2.6 Notasi Mol	18
Gambar 2.7 Notasi Pugar	19
Gambar 2.8 Penempatan Notasi Kres	19
Gambar 2.9 Penempatan Notasi Mol	19
Gambar 2.10 Nilai Nada	20
Gambar 2.11 Nilai Nada Notasi Balok	20
Gambar 2.12 Nilai Tanda Diam	21
Gambar 2.13 Nilai Tanda Diam Notasi Balok.....	21
Gambar 2.14 Paranada	23
Gambar 2.15 Penempatan Notasi pada Paranada	23
Gambar 2.16 Paranada	34
Gambar 2.17 Tanda Kunci G	25
Gambar 2.18 Tangga Nada Kres	27
Gambar 2.19 Tangga Nada Mol	28
Gambar 3.1 Diagram Konteks	35
Gambar 3.2 Diagram <i>Logic Level 0</i>	36
Gambar 3.3 Diagram <i>Logic Proses Cek Level 1</i>	37
Gambar 3.4 Diagram <i>Logic Proses Konversi Level 1</i>	37
Gambar 3.5 Tampilan Antar Muka Main Form	43
Gambar 3.6 Tampilan Antar Muka Form File	44
Gambar 3.7 Tampilan Antar Muka Form Notasi Angka	45
Gambar 3.8 Tampilan Antar muka Form Notasi Angka Generate	45

Gambar 3.9 Tampilan Antar Muka Form Notasi Balok Generate	46
Gambar 4.1 Main Form	48
Gambar 4.2 Toolbar Main Form	49
Gambar 4.3 Popup Menu Export Button	49
Gambar 4.4 Menu Generate Button	50
Gambar 4.5 Theme Gallery	50
Gambar 4.6 Form File	51
Gambar 4.7 Form Notasi Angka	52
Gambar 4.8 Form Notasi Angka Generate	53
Gambar 4.9 Form Notasi Balok Generate	54

DAFTAR TABEL

Tabel 2.1 Perubahan Nada Dasar Kres	26
Tabel 2.2 Perubahan Nada Dasar Mol	27
Tabel 4.1 Tabel Pengujian Tombol-Tombol Main Form	55
Tabel 4.2 Tabel Pengujian Form File	56
Tabel 4.3 Tabel Pengujian Tombol-Tombol Form Not Angka	56
Tabel 4.4 Tabel Pengujian Form Notasi Angka Generate	57
Tabel 4.5 Tabel Pengujian Form Notasi Balok	58

DAFTAR SIMBOL

Simbol	Keterangan
	Entitas Eksternal Menunjukkan Entitas yang menerima dan mendapatkan
	Proses Menunjukkan proses yang dilakukan
	Aliran Data Menunjukkan jalannya arus data
	Data Store Menunjukkan penyimpanan data

BAB 1

PENDAHULUAN

1.1 Latar Belakang Masalah

Perkembangan teknologi informasi memberikan manfaat tersendiri pada berbagai aspek kehidupan sehari-hari. Dengan hadirnya teknologi, berbagai pemenuhan hidup manusia menjadi lebih mudah. Pada dasarnya teknologi membawa pengaruh positif dalam kehidupan manusia, bahkan kemajuan teknologi menjadi bukti perkembangan kemampuan manusia untuk menggunakan pikirannya dalam mengelola potensi diri manusia itu sendiri.

Hadirnya teknologi perangkat lunak dibidang musik yang berkembang pesat, membawa dampak positif yang memudahkan seorang musisi maupun penikmat musik dalam membuat maupun mendengarkan musik. Saat ini perkembangan perangkat lunak dari perangkat lunak pengolah suara berkembang sampai perangkat lunak untuk pembuatan musik, pembelajaran *chord* alat musik, tangga nada, sampai pembuatan partitur.

Partitur merupakan dasar dalam pendidikan musik. Banyak orang yang memanfaatkan partitur dalam bermain musik. Contohnya dalam paduan suara, *orchestra*, *ensemble*, maupun pengajar dan siswa/siswi dalam kegiatan belajar mengajar seni musik.

Partitur ditulis dengan tanda dasar (*Clef*) yang di tempatkan pada garis paranada (*Staff*). Fungsi tanda dasar adalah untuk menunjukkan letak nada. Ada tiga macam tanda kunci pada partitur, yaitu kunci G (*Treble clef*), C (*Middle Clef*) dan F (*Bass Clef*). Setiap tanda kunci mempunyai kegunaan yang berbeda-beda dalam aplikasinya. Kunci G digunakan untuk menuliskan nada-nada tinggi, kunci F digunakan untuk menuliskan nada-nada rendah, kunci C digunakan untuk menuliskan nada-nada sedang.

Selain partitur dengan notasi balok, terdapat notasi angka yang dituliskan dengan angka-angka yang dimulai dari angka nol sampai dengan tujuh. Notasi balok dan notasi angka sangat berguna untuk dalam mendokumentasikan sebuah karya musik maupun dalam proses pembelajaran musik.

Banyak perangkat lunak yang menyediakan fitur dalam pembuatan notasi angka maupun notasi balok dan situs-situs yang memberikan pengetahuan dalam pembuatan partitur not balok, not angka, tangga nada maupun teori-teori dasar musik lainnya. Namun penulis ingin membuat semua hal itu dalam satu perangkat lunak agar pengguna dapat lebih mudah dalam pembuatan partitur dengan dasar notasi angka yang dapat melakukan konversi ke dalam partitur notasi balok.

Berdasarkan latar belakang tersebut, penulis menyimpulkan bahwa proyek tugas akhir yang akan dirancang adalah **“Perancangan Perangkat Lunak Konversi Notasi Angka ke Dalam Notasi Balok dengan Visual Basic”**.

1.2 Rumusan Masalah

Dari uraian latar belakang permasalahan di atas, maka hal-hal yang akan diteliti oleh penulis dirumuskan sebagai berikut:

1. Bagaimana pembangunan perangkat lunak partitur guna membantu pengguna dalam membuat notasi angka dan notasi balok dengan Visual Studio?
2. Bagaimana pengguna dapat menganalisa nada-nada yang telah disusun oleh pengguna?

1.3 Tujuan Penulisan

Tujuan penulisan ini adalah untuk memenuhi tugas akhir penulis. Selain itu juga penulis mempunyai tujuan untuk membuat suatu perangkat lunak penyusun partitur. Penjelasan nya adalah sebagai berikut :

1. Mempermudah seorang musisi (guru, seniman musik, dll) maupun siswa/siswi dalam pembuatan partitur dengan notasi angka dan notasi balok.
2. Mempermudah pembuatan partitur notasi angka dan notasi balok.
3. Mempermudah pembuatan notasi balok yang secara langsung terkonversi dari partitur notasi angka yang telah ditulis pengguna.

1.4 Batasan Masalah

Berdasarkan masalah-masalah yang telah dikemukakan di atas, pokok-pokok permasalahan yang menjadi batasan pembahasan penelitian ini supaya perancangan pembangunan sistem tidak menyimpang, maka diperlukan suatu batasan masalah sebagai acuan dalam penyelesaian masalah antara lain:

1. Perangkat lunak dapat menerima *input* angka dan tanda kres untuk mengkonversi secara otomatis ke dalam notasi balok.
2. Perangkat lunak mengkonversi notasi dalam tanda kunci G (*Treble Clef*).
3. Perangkat lunak dapat menerima tiga input birama, yaitu 3/4, 4/4 dan 6/8.
4. Perangkat lunak menerima masukan dua tempo, yaitu 60bpm dan 120bpm.

1.5 Kegunaan Hasil

Dari uraian yang telah disampaikan penulis, maka perangkat lunak yang dibangun penulis mempunyai kegunaan, sebagai berikut :

1. Menyusun partitur notasi angka
2. Menyusun partitur notasi balok secara otomatis dari partitur notasi angka yang telah ditulis oleh pengguna.
3. Sebagai media pembelajaran mengenai teori musik dasar, khususnya partitur.

1.6 Metodologi Penelitian

Dalam proses perancangan perangkat lunak, penulis menggunakan metode penelitian sebagai berikut :

1. Observasi, melakukan penelitian dan pembelajaran kepada dosen, musisi maupun pengajar musik untuk memperoleh data yang akurat dan update.
2. Studi literatur, mengunjungi situs-situs yang menyediakan informasi mengenai partitur dengan berbagai kunci dasar dan berbagai bahasa pemrograman yang tepat untuk digunakan penulis dalam membangun perangkat lunak.
3. Perancangan, dengan melakukan analisis perancangan perangkat lunak yang akan dirancang oleh penulis dari penggunaan bahasa pemrograman yang sesuai,

pemilihan perangkat lunak sebagai *database*, pembangunan objek-objek yang dibutuhkan, perancangan *interface* dan *coding*.

1.7 Sistematika Penulisan

Dalam proses penyusunan laporan, sistematika penulisan yang akan digunakan adalah sebagai berikut :

1. Bab 1 Pendahuluan

Bab ini digunakan untuk menjelaskan latar belakang, rumusan masalah, tujuan penulisan, batasan masalah, kegunaan hasil, metodologi penelitian dan sistematika penulisan.

2. Bab 2 Landasan Teori

Bab ini menjelaskan definisi-definisi dari teori secara umum mengenai perangkat lunak, pemrograman, teori dasar musik dan teori-teori lainnya yang berhubungan dengan tugas akhir penulis yang digunakan penulis dalam menyusun karya tulis ini.

3. Bab 3 Analisis dan Perancangan

Bab ini menjelaskan tentang analisis dan perancangan perangkat lunak oleh penulis berdasarkan pendekatan yang dipilih oleh penulis. Bab ini terdiri dari gambaran umum perangkat lunak, spesifikasi kebutuhan, diagram dan kemungkinan skenario yang terjadi.

4. Bab 4 Implementasi dan Pengujian Perangkat Lunak

Bab ini menjelaskan spesifikasi perangkat keras yang dibutuhkan dalam perangkat lunak yang dirancang serta pengujian komponen-komponen perangkat lunak yang dirancang.

5. Bab 5 Kesimpulan dan Saran

Bab ini berisi kesimpulan dan saran atas rumusan-rumusan masalah yang dibahas oleh penulis tentang perangkat lunak yang dirancang.

BAB 2

LANDASAN TEORI

2.1 Rekayasa Perangkat Lunak

Menurut Roger S. Pressman (Pressman, 2010:5), perangkat lunak dapat didefinisikan sebagai berikut :

1. Instruksi-instruksi (program komputer) yang apabila dieksekusi akan memberikan hasil yang diharapkan.
2. Struktur data yang memungkinkan program dapat melakukan manipulasi terhadap informasi.
3. Dokumen yang menjelaskan informasi serta penggunaan program.

Perangkat lunak memiliki beberapa karakteristik. Menurut Roger S. Pressman (Pressman, 2010:4-6), karakteristik perangkat lunak yaitu:

1. Perangkat lunak dibangun dan dikembangkan, tidak dibuat dalam bentuk fisik.
2. Perangkat lunak tidak pernah usang.

Sebagian perangkat lunak dirancang dengan *custom-built*, serta dapat dirakit dari komponen yang sudah ada.

Rekayasa atau teknik adalah penerapan ilmu dan teknologi untuk menyelesaikan permasalahan manusia. (Simarmata, 2010:10). Hal-hal tersebut diselesaikan melalui ilmu, pengetahuan dan pengalaman praktis yang diterapkan untuk merancang obyek ataupun proses yang berguna.

Menurut Janner Simarmata dalam bukunya Rekayasa Perangkat Lunak, rekayasa perangkat lunak adalah satu bidang profesi yang mendalami cara-cara pengembangan perangkat lunak termasuk pembuatan, pemeliharaan, manajemen organisasi pengembangan perangkat lunak, dan sebagainya.

Ada beberapa pengertian rekayasa perangkat lunak menurut beberapa ahli, diantaranya:

1. Rekayasa perangkat lunak Menurut Stephen R. Schach

Rekayasa perangkat lunak adalah sebuah disiplin dimana dalam menghasilkan perangkat lunak bebas dari kesalahan dan dalam pengiriman anggaran tepat waktu serta memuaskan keinginan pemakai.

2. Rekayasa perangkat lunak Menurut Fritz Bauer

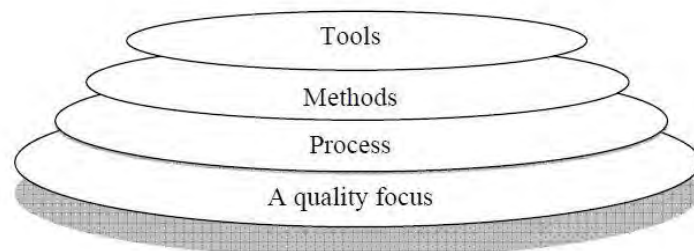
Rekayasa perangkat lunak adalah penetapan dan penggunaan prinsip rekayasa dalam rangka memperoleh perangkat lunak yang dapat dipercaya dan dapat bekerja secara efisien pada mesin nyata.

3. Rekayasa perangkat lunak menurut *Institute of Electrical and Electronic Engineers* (IEEE)

Rekayasa perangkat lunak adalah sebuah studi pendekatan dan aplikasi secara sistematis, disiplin pengembangan operasi dan pemeliharaan perangkat lunak yang kesemuanya itu merupakan aplikasi rekayasa yang berkaitan dengan perangkat lunak.

Dari beberapa pengertian di atas, dapat disimpulkan bahwa rekayasa perangkat lunak adalah suatu bidang profesi atau disiplin ilmu yang menerapkan teori-teori, prinsi-prinsip dan ilmu-ilmu tertentu daam merancang maupun mengembangkan suatu perangkat lunak yang dapat berfungsi dengan baik.

Rekayasa perangkat lunak merupakan sebuah disiplin yang terdiri dari empat lapisan proses, yaitu: alat bantu (*tools*), metode (*method*), proses (*process*), fokus kualitas (*quality of focus*).



Gambar 2.1 : *Software Engineering Layer*

Sumber : (Pressman, 2010:14)

Seperti halnya daerah rekayasa yang lain, rekayasa perangkat lunak digabungkan melalui elemen kunci, yaitu:

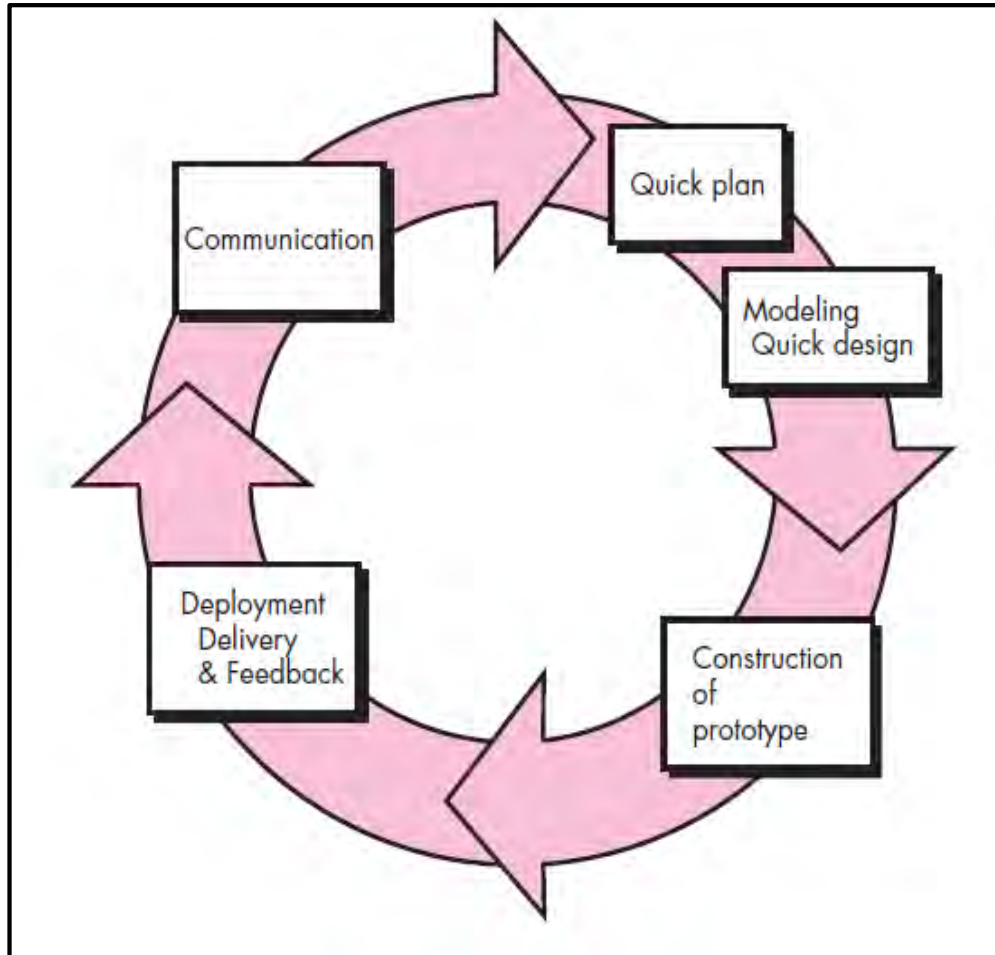
1. Kelayakan : menemukan solusi yang baik untuk menyelesaikan masalah.
2. Maksimalisasi nilai : memaksimalkan nilai dari solusi yang disediakan.
3. Strategi yang efektif : mengadopsi suatu strategi yang efektif untuk mengembangkan solusi.
4. Pemodelan : perancangan suatu gambaran visual dari tahap sebelum solusi sampai implementasinya.

2.2 Model *Prototype*

Pada model *prototype*, pengembang lebih sering mendefinisikan suatu tujuan yang umum untuk perangkat lunak, tetapi tidak mengidentifikasi persyaratan yang rinci untuk fungsi-fungsi dan fitur perangkat lunak tersebut. (Pressman, 2010:43). Model *prototyping* merupakan model proses yang berdiri sendiri. Model *prototyping* mengimplementasikan *prototyping paradigm* oleh pengembang untuk membantunya lebih memahami perangkat lunak yang akan dibangun ketika persyaratan-persyaratan yang dibutuhkan atau diperlukan tidak jelas.

Prototyping paradigm dimulai dengan komunikasi yang bertujuan untuk menentukan tujuan keseluruhan untuk sebuah perangkat lunak yang akan dirancang, mengidentifikasi kebutuhan yang diperlukan dalam perancangan perangkat lunak dan *outline* perangkat lunak. *Prototyping* merupakan perencanaan dan pemodelan yang cepat dalam sebuah *quick design*. Sebuah *quick design* berfokus pada representasi dari aspek-aspek perangkat lunak pada pemakai dan mengacu pada pembuatan *prototype*. *Prototype* yang digunakan dan dievaluasi oleh *stackholder* yang memberikan umpan balik adalah untuk memperbaiki persyaratan-persyaratan dan kebutuhan perangkat lunak yang dibutuhkan oleh pengembang.

Prototype berfungsi sebagai sebuah mekanisme untuk mengidentifikasi kebutuhan perangkat lunak. Jika *prototype* akan dibangun, pengembang dapat menggunakan fragmen program yang tersedia ataupun menerapkan alat-alat yang memungkinkan perancangan perangkat lunak secara cepat.



Gambar 2.2 : Model Prototype

Sumber : (Presman, 2010:43)

Prototype bisa berfungsi sebagai sistem awal, meskipun beberapa *prototype* awal tidak digunakan atau dapat dikatakan dibangun sebagai contoh dasar dan berkembang secara perlahan menjadi suatu perangkat lunak akhir. Hal ini menunjukkan bahwa *prototype* perlahan berkembang menjadi sistem yang sebenarnya. Namun, *prototyping* dapat menjadi masalah karena alasan berikut :

1. *Stakeholder* yang terlibat dalam perancangan perangkat lunak yang diinginkan pengguna tidak menyadari keadaan atau waktu yang singkat dalam perancangan perangkat lunak dan perangkat lunak yang dibangun belum mencapai kualitas perangkat lunak tertentu dan lama pemeliharaan perangkat lunak tersebut.
2. Sebuah sistem operasi, bahasa pemrograman yang digunakan hanya sebuah sistem operasi dan bahasa pemrograman yang tersedia dan dikenal. Selain itu algoritma yang digunakan tidak efisien. Hal ini menjadi kurang dari ideal. Pilihan pengembang tersebut telah menjadi bagian dari sistem.

Meskipun masalah dapat terjadi, *prototyping* dapat menjadi model yang efektif dalam merancang perangkat lunak. Inti dari model *prototype* adalah menentukan aturan-aturan dan syarat-syarat dasar pada awal pengembangan yaitu, semua *stackholder* harus setuju bahwa *prototype* yang akan dibangun berfungsi sebagai mekanisme untuk mendefinisikan persyaratan.

2.3 Pemrograman Terstruktur

Pemrograman terstruktur adalah suatu proses untuk mengimplementasikan urutan langkah untuk menyelesaikan suatu masalah dalam bentuk program (Sugiyono, 2005:17). Selain pengertian diatas pemrograman terstruktur adalah suatu aktifitas pemrograman dengan memperhatikan urutan langkah-langkah perintah secara sistematis, logis, dan tersusun berdasarkan algoritma yang sederhana dan mudah dipahami. Prinsip dari pemrograman terstruktur adalah Jika suatu proses telah sampai pada suatu titik atau langkah tertentu, maka proses selanjutnya tidak boleh mengeksekusi langkah sebelumnya atau kembali lagi ke baris sebelumnya, kecuali pada langkah-langkah untuk proses berulang (*Loop*).

Pemrograman terstruktur memiliki beberapa sifat atau karakteristik tertentu. Sifat-sifat pemrograman terstruktur yaitu:

1. Memuat teknik pemecahan masalah yang logis dan sistematis.
2. Memuat algoritma yang efisien, efektif dan sederhana.
3. Program disusun dengan logika yang mudah dipahami.

4. Tidak menggunakan perintah *GOTO*.
5. Biaya pengujian program relatif rendah.
6. Memiliki dokumentasi yang baik.
7. Biaya perawatan dan dokumentasi yang dibutuhkan relatif rendah.

Ada perbedaan dari pemrograman terstruktur dan pemrograman berorientasi objek. Pemrograman berorientasi objek (*Object-Oriented Programming* atau *OOP*) merupakan paradigma pemrograman yang berorientasikan kepada objek. Semua data dan fungsi di dalam paradigma ini dibungkus dalam kelas-kelas atau objek-objek. Setiap objek dapat menerima pesan, memproses data, dan mengirim pesan ke objek lainnya (Binanto,2005:11). Sedangkan dalam pemrograman terstruktur adalah kebalikan dari pemrograman berorientasi objek yaitu sebuah cara pemrosesan data yang terstruktur dalam analisa, cara dan penulisan pemrograman. Karena pemrograman yang terstruktur, dalam pembuatannya antara satu *line* pemrograman dengan yang lainnya berhubungan.

Pemrograman Terstruktur terdiri dari pemecahan masalah yang besar menjadi masalah yang lebih kecil dan seterusnya, sedangkan untuk pemrograman berorientasi objek terdiri dari pengkelompokan kode dengan data yang mana setiap objek berfungsi secara independen sehingga untuk setiap perubahan kode tidak tergantung pada kode yang lainnya atau lebih dikenal dengan modular. Terdapat juga perbedaan secara spesifik antara pemrograman berorientasi objek dengan pemrograman terstruktur, yaitu pada kelas dan objek. Pada pemrograman terstruktur tidak terdapat kelas dan objek. Pemrograman terstruktur memiliki keunggulan dalam melakukan pemrograman sederhana dan efisien tetapi permodelan ini lebih susah untuk dipahami oleh orang-orang selain pembuat program itu sendiri.

Perancangan Terstruktur (*Structured Analysis and Design* atau *SSAD*) adalah Metode ini diperkenalkan pada tahun 1970, yang merupakan hasil turunan dari pemrograman terstruktur. Metode pengembangan dengan metode terstruktur ini terus diperbaiki sampai akhirnya dapat digunakan dalam dunia nyata. Perancangan ini bertujuan untuk membuat model solusi terhadap masalah yang sudah dimodelkan secara lengkap pada tahap analisis terstruktur.

Terdapat kelebihan dalam perancangan terstruktur. Kelebihan tersebut diantaranya yaitu:

1. *Milestone* diperlihatkan dengan jelas yang memudahkan dalam manajemen proyek
2. Merupakan pendekatan visual, ini membuat metode ini mudah dimengerti oleh pengguna atau programmer.
3. Penggunaan analisis grafis dan tool seperti *data flow diagram*.
4. Merupakan metode yang diketahui secara umum pada berbagai industry.
5. Sudah diterapkan begitu lama sehingga metode ini sudah matang dan layak untuk digunakan.
6. Memungkinkan untuk melakukan validasi antara berbagai kebutuhan
7. Relatif simpel dan mudah dimengerti.

Terdapat kekurangan dalam perancangan terstruktur. Kekurangan tersebut diantaranya yaitu:

1. Berorientasi utama pada proses, sehingga mengabaikan kebutuhan non-fungsional.
2. Prinsip dasarnya merupakan pengembangan *non-iterative (waterfall)*, akan tetapi kebutuhan akan berubah pada setiap proses.
3. Interaksi antara analisis atau pengguna tidak komprehensif, karena sistem telah didefinisikan dari awal, sehingga tidak adaptif terhadap perubahan (kebutuhan-kebutuhan baru).
4. Selain dengan menggunakan desain logic dan *data flow diagram*, tidak cukup *tool* yang digunakan untuk mengkomunikasikan dengan pengguna, sehingga sangat sulit bagi pengguna untuk melakukan evaluasi.

2.4 Diagram

Terdapat diagram-diagram atau bagan untuk menjelaskan arus data ataupun arus kerja dalam pemrograman terstruktur. Berikut adalah penjelasannya.

2.4.1 Data Flow Diagram (DFD)

Data flow diagram sering digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir (Kadir,2009:34). *Data flow diagram* merupakan alat yang digunakan pada metodologi pengembangan sistem yang terstruktur dan dapat mengembangkan arus data di dalam sistem dengan terstruktur dan jelas.

Ketentuan Pembuatan *data flow diagram* adalah sebagai berikut :

1. Setiap lingkaran proses minimal mempunyai 1 *input* dan 1 *output*.
2. Antara entitas eksternal dan entitas eksternal lainnya tidak berhubungan langsung tanpa adanya proses.
3. Antara entitas eksternal dan penyimpanan data tidak berhubungan.
4. Antara 2 penyimpanan data atau data *store* tidak berhubungan langsung tanpa adanya proses.
5. Satu arus data tidak dapat berarti dua nama arus data.
6. Setiap aliran data (*data flow*) harus mempunyai nama atau label yang bermakna.
7. Nama arus data tidak menggambarkan suatu proses.

Simbol-simbol yang digunakan dalam *data flow diagram* adalah entitas, proses, arus data dan *data store*. Penjelasan sebagai berikut:

1. Entitas

Entitas atau kesatuan luar merupakan kesatuan di lingkungan luar sistem yang dapat berupa orang, organisasi atau sistem lainnya yang berada di lingkungan luarnya yang akan memberikan *input* atau menerima *output* dari sistem. Entitas disimbolkan dengan notasi kotak.

2. Proses

Suatu proses adalah kegiatan yang dilakukan oleh orang, mesin, atau komputer dari hasil suatu arus data yang masuk ke dalam proses untuk menghasilkan arus data yang akan keluar dari proses. Simbol proses berupa lingkaran atau persegi panjang bersudut tumpul.

3. *Arus Data*

Arus Data (*data flow*) pada *data flow diagram* diberi simbol panah. Arus data ini mengalir di antara proses, simpanan data (*data store*) dan kesatuan luar (entitas eksternal). Arus data ini menunjukkan arus data yang dapat berupa masukan untuk sistem atau hasil dari proses sistem.

4. *Data Store*

Data Store merupakan simpanan dari data yang dapat berupa hal-hal sebagai berikut, sebagai gambaran suatu *file* atau *database* di sistem komputer, sebagai gambaran suatu arsip atau catatan manual, sebagai gambaran suatu kotak tempat data di meja seseorang dan sebagai gambaran suatu tabel acuan manual.

Simpanan data pada *data flow diagram* disimbolkan dengan sepasang garis horizontal paralel yang tertutup di salah satu ujungnya.

Metode pembuatan *data flow diagram* dimulai dengan rancangan sistem secara garis besar, lalu memecahnya menjadi bagian-bagian yang lebih rinci. Jadi, mula-mula akan digambarkan dulu diagram konteks, lalu *data flow diagram* Level nol (baik diagram fisik atau lojik), lalu *data flow diagram* level satu dan seterusnya hingga suatu proses tidak bisa dirinci lagi.

Context Diagram merupakan langkah pertama dalam pembuatan *data flow diagram*. Pada context diagram dijelaskan sistem apa yang dibuat dan entitas eksternal apa saja yang terlibat. Dalam *context diagram* harus ada arus data yang masuk dan arus data yang keluar.

Logic diagram yaitu representasi grafik dari suatu sistem yang menggambarkan proses-proses dari sistem tersebut berikut aliran-aliran datanya. *Logic diagram* menunjukkan apa yang dilakukan oleh suatu sistem, tanpa menspesifikasikan dimana, bagaimana, dan oleh siapa proses-proses dalam sistem dilakukan. Keuntungan *logic diagram* dibandingkan dengan diagram fisik adalah kita dapat memusatkan perhatian pada fungsi-fungsi yang dilakukan oleh sistem.

2.4.2 Spesifikasi Proses

Pembuatan spesifikasi proses digunakan untuk memastikan bahwa setiap proses memiliki semua *input* yang diperlukan untuk menghasilkan *output* yang sesuai dengan kebutuhan. Spesifikasi proses memiliki tujuan, yaitu:

1. Mengurangi ambiguitas dari sebuah proses.
2. Untuk memperoleh deskripsi yang tepat mengenai apa yang ingin dicapai. Biasanya tergabung dengan paket spesifik yang diberikan pada programmer.
3. Memvalidasi desain sistem yang telah dilakukan sebelumnya.

Terdapat beberapa kelompok proses yang tidak perlu membuat spesifikasi proses, yaitu:

1. Proses-proses yang menggambarkan input atau output fisik, seperti baca dan tulis data. Proses-proses seperti ini biasanya hanya memerlukan logika yang sederhana.
2. Proses-proses yang menggambarkan validasi data yang sederhana. Spesifikasi proses boleh dibuat untuk proses editing yang kompleks.
3. Proses-proses yang menggunakan kode yang telah ada sebelumnya, seperti subprogram atau fungsi.

Terdapat format-format yang ditentukan dalam pembuatan spesifikasi proses. Format dalam pembuatan spesifikasi proses tersebut yaitu:

1. Yang diambil menjadi spesifikasi proses adalah level yang terbesar dari sebuah *data flow diagram*.
2. Nomor proses harus sama dengan nomor proses di *data flow diagram*.
3. Nama proses harus sama dengan nama proses di *data flow diagram*.
4. Deskripsi singkat dari apa yang ingin dicapai oleh proses.
5. Daftar aliran input data, menggunakan nama yang sama pada *data flow diagram*. Nama data-data yang akan digunakan harus sama dengan yang ada di kamus data untuk memastikan konsistensi dan komunikasi yang baik.
6. Daftar aliran data output, menggunakan nama yang sama dengan yang ada pada *data flow diagram*.
7. Adanya indikasi dari tipe sebuah proses : *batch*, *online*, atau *manual*.

8. Jika sebuah proses menggunakan kode program yang telah ada sebelumnya, maka sertakan nama *subprogram* atau fungsi yang mengandung kode program tersebut.
9. Sebuah deskripsi dari logika proses yang menyatakan kebijakan atau aturan bisnis dalam bahasa sehari-hari, bukan dalam bahasa komputer (*pseudocode*).
10. Cantumkan daftar masalah-masalah yang belum terselesaikan, bagian logik yang belum selesai, atau masalah-masalah lainnya.

Terdapat tiga jenis dalam pembuatan spesifikasi proses. Jenis-jenis tersebut ialah *structured english*, *desicion table* dan *desicion tree*.

2.5 Notasi Musik

Dalam seni musik, nada-nada dilambangkan dengan not-not. Setiap nada dilambangkan dengan satu not tertentu. Not-not inilah yang digunakan oleh musisi untuk mendokumentasikan karya-karyanya sehingga dapat kita baca atau mainkan.

Notasi merupakan cara yang digunakan oleh musisi sebagai alat untuk mendokumentasi karyanya. Notasi juga adalah alat bagi para penikmat musik untuk membaca karya musik yang dihasilkan oleh musisi tersebut. Selain itu notasi merupakan teks musik untuk pedoman bagi pemain musik dalam bermain musik secara berkelompok agar lebih harmonis dengan komposisi yang baik.

Menurut Mark McGrain dalam bukunya *Music Notation: Theory and Technique for Music*, *Notation symbols, when properly applied, decribed complex musical sound in the simplest, most concice manner possible.*

Menurut Pono Banoe dalam bukunya *Kamus Musik*, notasi adalah tulisan musik atau lambang musik. (Banoe, 2005:299).

Dari pengertian-pengertian notasi diatas, dapat disimpulkan bahwa notasi merupakan suatu alat atau media tulis yang ditulis dengan lambang-lambang tetentu guna mendokumentasi, membaca, memainkan karya musik agar lebih harmonis.

Secara umum, kita mengenal dua macam notasi, yakni notasi notasi angka dan notasi balok. Notasi yang banyak digunakan dalam seni musik daerah adalah notasi angka atau sistem balungan. Sementara notasi balok umumnya digunakan dalam seni musik modern.

1. Notasi Angka

Notasi angka menggunakan angka-angka untuk merujuk pada satu nada. Berawal dari angka 1 (Do) hingga angka 7 (Si).

1	2	3	4	5	6	7
Do	Re	Mi	Fa	Sol	La	Si

Untuk nada-nada yang lebih tinggi dari nada-nada dasar, maka digunakan simbol titik yang terletak pada bagian atas notasi angka. Untuk nada-nada yang lebih rendah dari nada-nada dasar, maka digunakan simbol titik yang terletak pada bagian bawah notasi angka.

2. Notasi Balok

Notasi balok ditulis pada garis balok not atau sangkar nada (paranada). Pada notasi ini menggunakan lambang notasi yang berbeda. Penulisan notasi balok menggunakan lambing bulat dengan tiang dan bendera. Notasi ini lebih sering digunakan dalam musik modern. Alasannya adalah notasi balok ini memiliki nada dengan tinggi dan letak yang standar.

2.6 Elemen Notasi Musik

Terdapat elemen-elemen dalam penulisan notasi angka maupun notasi balok. Elemen-elemen tersebut ialah :

2.6.1 Lambang Notasi Musik

Dalam penulisan notasi terdapat lambang-lambang yang digunakan. Pada notasi angka, lambang yang digunakan adalah angka satu sampai dengan tujuh yang mewakili satu nada.

Dalam penulisan notasi balok, lambang yang digunakan adalah kepala not (bulat), tiang not dan bendera not. Kepala not dapat tertutup maupun terbuka. Hal ini membedakan panjang nada atau nilai nada pada notasi.

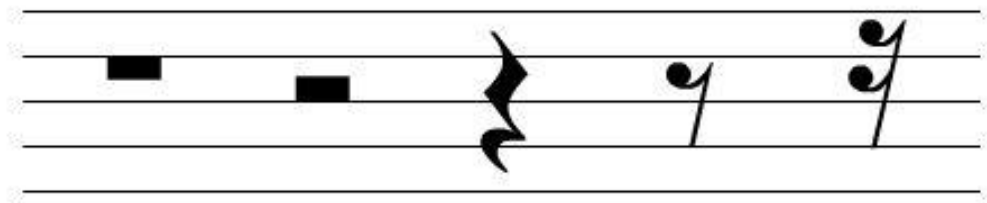


Gambar 2.3 : Notasi Balok

Sumber : (Mulyanto, 2007:4)

2.6.2 Tanda Diam

Selain lambang untuk penulisan not, terdapat pula lambang untuk menyatakan tanda berhenti atau diam. Lambang ini berfungsi untuk memberi jeda dalam membaca notasi. Pada notasi angka, lambang diam dituliskan dengan lambang 0 (nol) dan diberikan penambahan titik untuk menambahkan nilai notasi atau nilai ketuk dari tanda diam tersebut. Pada notasi balok, penulisan lambang diam adalah sebagai berikut



Gambar 2.4 : Tanda Diam

Sumber : (Mulyanto, 2007:5)

2.6.3. Tanda Kromatis (*Accidentals*)

Tanda kromatis atau *accidentals* adalah tanda atau simbol-simbol yang digunakan menaikkan atau menurunkan nada. Yang merupakan tanda kromatis adalah kres, mol dan pugar.

1. Tanda Kres (*Sharp*)

Tanda kres digunakan untuk menaikkan nada sebanyak $\frac{1}{2}$ laras. Tanda kres ditulis dengan lambang .



Gambar 2.5 : Notasi Kres

Sumber : (Mulyanto, 2007:8)

2. Tanda Mol (*Flat*)

Tanda mol digunakan untuk menurunkan nada sebanyak $\frac{1}{2}$ laras. Tanda mol ditulis dengan lambang .



Gambar 2.6 : Notasi Mol

Sumber : (Mulyanto, 2007:8)

3. Tanda Pugar (*Neutral*)

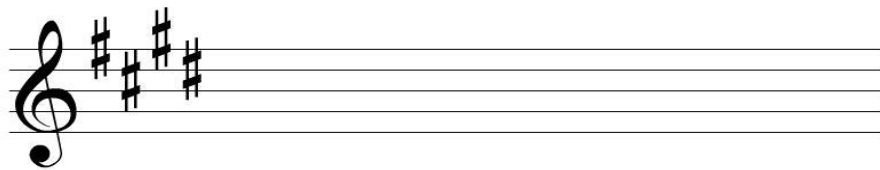
Tanda pugar berfungsi untuk mengembalikan nada yang sebelumnya dikenai tanda kres atau mol ke nada awal. Tanda pugar ditulis dengan lambang .



Gambar 2.7 : Notasi Pugar

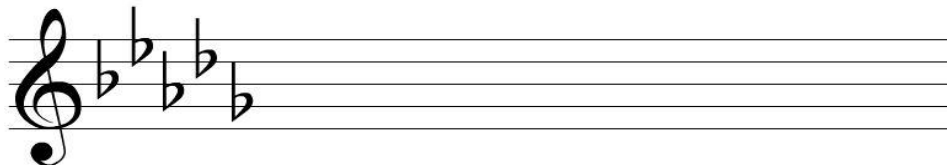
Sumber : (Mulyanto, 2007:8)

Tanda kres dan mol berlaku pada bar ketika simbol tersebut muncul. Pada bar berikutnya, tanda kres dan mol tidak berlaku kembali, kecuali dihubungkan dengan suatu garis lengkung yang disebut *tie*.



Gambar 2.8 : Penempatan Notasi Kres

Sumber : (Mulyanto, 2007:9)








Gambar 2.9 : Penempatan Notasi Mol

Sumber : (Mulyanto, 2007:9)

2.7 Nilai Nada

Nilai nada merujuk pada panjang nada yang dimainkan. Nilai nada pula merujuk pada banyaknya ketukan yang dimainkan oleh nada tersebut. Berikut adalah nilai nada beserta bentuk notasi balok.

Bentuk Not	Nama USA	Nama UK	Nama Ind	Nilai Not
	whole	Semibreve	Penuh	4 ketuk
	half	minim	1/2	2 ketuk
	quarter	crotchet	1/4	1 ketuk
	eight	quaver	1/8	1/2 ketuk
	sixteen	semiquaver	1/16	1/4 ketuk













Gambar 2.10 : Nilai Nada

Sumber : (Nickol, 2005:5)

Selain itu, terdapat lambang titik yang berada pada bagian belakang nada. Lambang ini berfungsi memperpanjang nada yang berada di depannya. Jika ada dua titik pada bagian belakang nada, maka titik terakhir bernilai setengah dari lambang titik di depannya. Ada pula titik yang ditempatkan pada bagian depan nada. Hal ini bertujuan menggenapi panjang nada sehingga sesuai dengan panjang nada lainnya dalam satu birama. Pada notasi angka penulisan tanda titik sebagai berikut

| 5 . . . | 1 . 7 . | 1 . 7 . | 5 . . . |







Pada notasi balok penulisan tanda titik diletakan pada bagian belakang lambang notasi balok. Penjelasan sebagai berikut sebagai berikut :

Bentuk not bertitik	Nilai
	
	
	
	
	
	

Gambar 2.11 : Nilai Nada Notasi Balok




























Sumber : (Nickol, 2005:6)

Lambang titik pun terdapat pada lambang diam. Fungsi lambang titik pada lambang diam pun sama seperti pada lambang nada.

BENTUK TANDA DIAM	NAMA TANDA DIAM	NILAI TANDA DIAM
	Tanda diam Penuh	4 ketuk
	Tanda diam Setengah	2 ketuk
	Tanda diam Segerempot	1 ketuk
	Tanda diam Segendelapan	$\frac{1}{2}$ ketuk
	Tanda diam Segerenam belas	$\frac{1}{4}$ ketuk
	Tanda diam Segertiga puluh dua	$\frac{1}{8}$ ketuk

Gambar 2.12 : Nilai Tanda Diam

Sumber : (Nickol, 2005:7)

Bentuk Tanda Diam	Nilai
	 + 
	 + 
	 + 
	 + 
	 + 
	 +  + 
	 +  + 
	 +  + 

Gambar 2.13 : Nilai Tanda Diam Notasi Balok

Sumber : (Nickol, 2005:7)

2.8 Birama dan Irama

Dalam penulisan notasi balok maupun notasi angka terdapat irama dan birama. irama berkaitan erat dengan ketukan, birama (*Metrum*) dan tempo. Irama adalah alunan nada-nada dalam lagu yang dimainkan secara teratur dan membentuk sebuah pola tertentu. (Ali, 2006:45).

Birama adalah ruangan yang dibatasi oleh dua garis vertikal sejajar yang di dalamnya terdapat sejumlah ketukan yang teratur dengan hitungan yang tetap. Hitungan tersebut akan berulang pada ketukan berikutnya.

Dalam penulisan notasi, setiap birama dibatasi dengan garis birama. Garis birama adalah garis tegak lurus yang menghubungkan garis kelima (atas) dan garis pertama (bawah) pada notasi. Dalam tiap birama terdapat ketukan yang sama. Ruang yang dibatasi oleh garis birama adalah bar.

Tanda birama atau metrum digunakan untuk merujuk pada banyaknya ketukan yang terdapat pada satu birama serta nilai nada dalam satu ketukan. penulisan birama pada notasi angka adalah dengan pecahan, contoh $2/4$ adalah birama yang di dalam satu birama tersebut terdapat dua ketukan dengan nilai masing-masing $1/4$. Penulisan birama pada notasi balok sebagai berikut

Selain itu pula garis vertikal yang membatasi pada birama memiliki fungsi yang lain, yaitu tanda pengulangan, tanda berhenti dan tanda perulangan pada bar sebelumnya. Berikut adalah birama yang umum digunakan dalam notasi musik :

1. Birama $2/4$

Di dalam satu birama, terdapat 2 ketukan dengan nilai not masing-masing $1/4$

| 1 2 | 1 2 | 1 2 | 1 2 |

2. Birama $3/4$

Di dalam satu birama, terdapat 3 ketukan dengan nilai not masing-masing $1/4$

| 1 2 3 | 1 2 3 | 1 2 3 | 1 2 3 |

3. Birama $4/4$

Di dalam satu birama, terdapat 4 ketukan dengan nilai not masing-masing $1/4$

| 1 2 3 4 | 1 2 3 4 | 1 2 3 4 | 1 2 3 4 |

4. Birama $6/8$

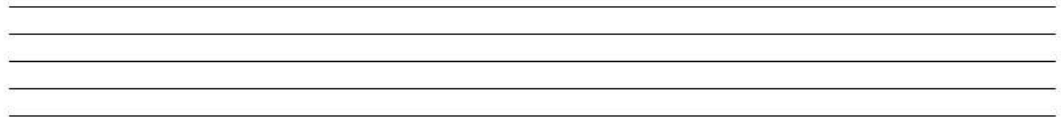
Di dalam satu birama, terdapat 6 ketukan dengan nilai not masing-masing $1/8$

| 1 2 3 1 2 3 | 1 2 3 1 2 3 | 1 2 3 1 2 3 | 1 2 3
1 2 3 |

2.9. Sangkar Nada atau Paranada

Sangkar nada atau paranada adalah lima garis sejajar dengan panjang yang sama dan memiliki empat spasi yang sama jaraknya dan untuk meletakkan notasi balok. (Tyas, 2006:82)

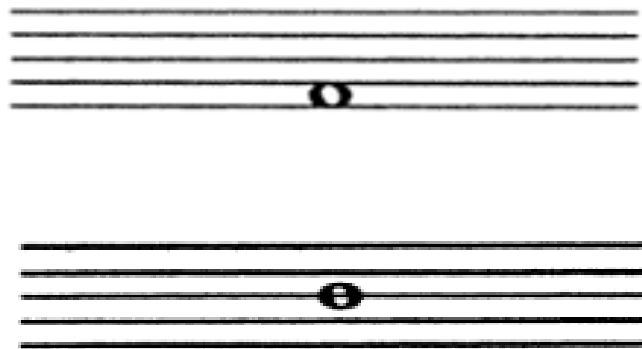
Sangkar nada atau paranada adalah lima garis horizontal untuk menunjukkan tinggi atau rendahnya nada. (Nickol, 2005:3)



Gambar 2.14 : Paranada

Sumber : (Aklat, 2014:16)

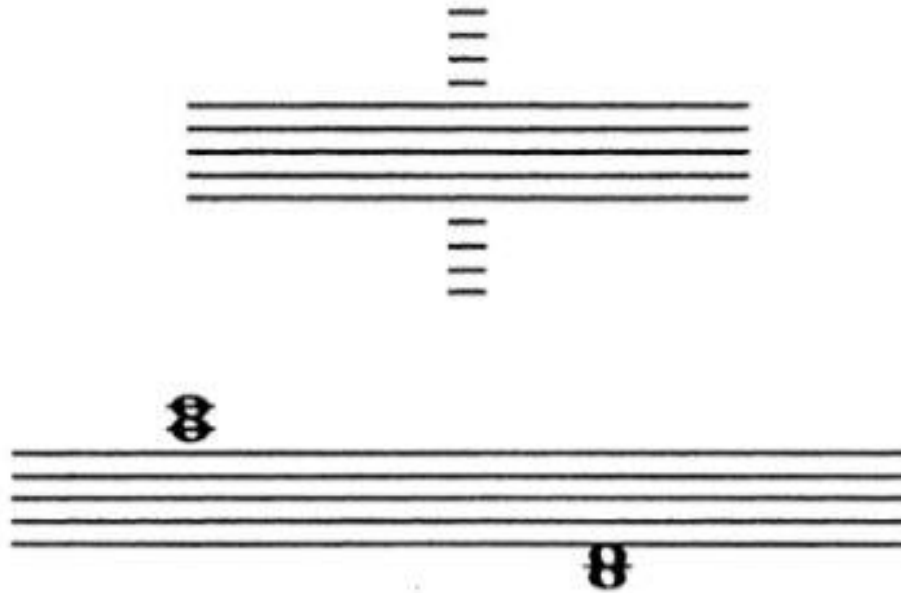
Not dapat ditulis dalam garis maupun pada ruang kosong antar garis. Setiap ruang memiliki arti yang berbeda. Penulisannya sebagai berikut



Gambar 2.15 : Penempatan Notasi pada Paranada

Sumber (Nickol, 2005:9)

Sangkar nada atau paranada dapat di perluas dengan garis bantu untuk menambah batasan tinggi atau rendahnya nada yang ditempatkan. Penulisannya sebagai berikut



Gambar 2.16 : Garis Paranada

Sumber : (Nickol, 2005:9)

2.10 Tanda Kunci

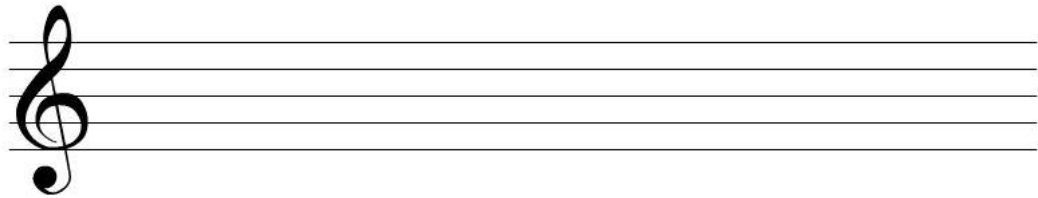
Tanda kunci adalah tanda yang terletak di sebelah kiri sangkar nada yang berfungsi sebagai penunjuk kunci nada yang digunakan.

Menurut Peter Nickol dalam bukunya *Panduan Praktis Membaca Notasi Musik*, tanda kunci adalah tanda yang digunakan untuk memberi identitas yang pasti pada not-not yang diletakan pada garis paradana. (Nickol, 2005:4).

Tanda kunci yang digunakan ada tiga, yaitu : G (*Treble Clef*), F (*Bass Clef*) dan C (*Middle Clef*).

1. Kunci G (*Treble Clef*)

Kunci G digambarkan dengan titik awal dari garis ke dua paranada. Kunci ini banyak digunakan dalam notasi balok. Kunci ini merupakan tempat bagi nada-nada dengan *range* paling atas. Alat musik yang biasa menggunakan notasi balok dengan kunci ini diantara lain adalah biola dan gitar.



Gambar 2.17 : Tanda Kunci G

Sumber : (Mulyanto, 2007:21)

2. Kunci F (*Bass Clef*)

Kunci F digambarkan dengan titik awal dari garis ke empat paranada. Kunci F biasanya digunakan untuk menyanyikan nada-nada dengan *range* rendah atau paling bawah. Alat musik yang biasa menggunakan notasi balok dengan kunci ini diantara lain adalah cello. Nada F terletak diantara titik dua yang juga merupakan titik yang merupakan titik awal gambar kunci ini.

3. Kunci C (*Middle Clef*)

Kunci C bisa berubah-ubah letaknya tetapi berbentuk tetap. Hal ini terjadi karena kunci ini digunakan untuk nada-nada dalam *range* tengah. Alat musik yang biasa menggunakan notasi balok dengan kunci ini diantara lain adalah viola.

Kunci C terletak pada garis ke dua. Berikut adalah letak nada berdasarkan tanda kunci dasar :

2.11 Perubahan Nada Dasar

Komposisi sebuah lagu pasti memiliki variasi nada dasar yang berbeda. Hal ini dilakukan biasanya untuk penyesuaian terhadap suara vokal manusia yang berbeda-beda. Penyesuaian ini dilakukan agar penyanyi dapat menyuarakan vokal dengan pas (tidak terlalu tinggi dan tidak terlalu rendah).

Perubahan nada dasar ditandai dengan jumlah lambang kres (*Sharp*) dan lambang mol (*Flat*).

1. Tangga Nada Berkres

Perubahan tangga nada ini diambil dari nada tingkat kelima (*kwint*) dari tangga nada sebelumnya yang dijadikan sebagai nada dasar yang baru. Nada ketujuh (*septime*) dari tangga nada yang baru selalu dinaikan sebanyak setengah laras. Penyusunannya sebagai berikut:

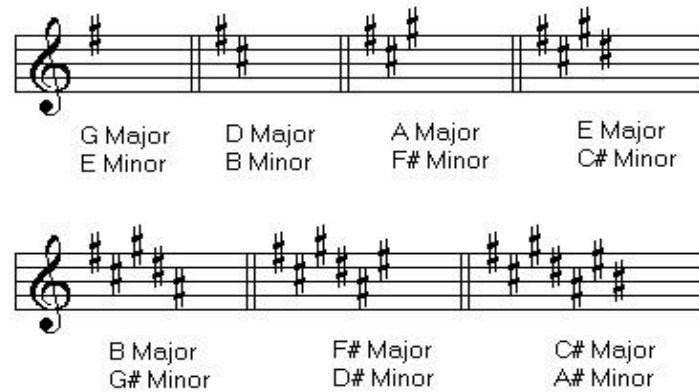
C = Do	C	D	E	F	G	A	B	C'
G = Do	G	A	B	C'	D'	E'	Fis	G'
D = Do	D	E	Fis	G	A	B	Cis'	D'

Dan seterusnya. Berikut adalah tabel perubahan tangga nada berkres:

Not mutlak	c	d	e	f	g	a	b	c'
Not angka	1	2	3	4	5	6	7	1̇
Jarak		1	1	1/2	1	1	1	1/2
1#	g	a	b	c'	d'	e'	fis'	g'
2#	d	e	fis	g	a	b	cis'	d'
3#	a	b	cis'	d'	e'	fis'	gis'	a'
4#	e	fis	gis	a	b	cis'	dis'	e'
5#	b	cis'	dis'	e'	fis'	gis'	ais'	b'
6#	fis	gis	ais	b'	cis'	dis'	eis'	fis'
7#	cis	dis	eis	fis	gis	ais	bis	cis'

Tabel 2.1 Perubahan Nada Dasar

Dalam paranada, lambang kres digunakan untuk menunjuk nada-nada mana saja yang dinaikan setengah laras. Lambang kres tersebut dituliskan hanya pada permulaan paranada setelah tanda kunci. Berikut adalah contoh penulisan lambang kres pada sangkar nada atau paranada:



Gambar 2.18 : Tangga Nada Kres

Sumber : (Surmani, 2004:33)

2. Tangga Nada Bermol

Perubahan tangga nada ini diambil dari tingkat keempat (*kwart*) dari tangga nada sebelumnya. Nada keempat dari tangga nada yang baru selalu diturunkan sebanyak setengah laras agar intervalnya tetap sama. Penyusunannya sebagai berikut:

C = Do C D E F G A B C'
 F = Do F G A Bes C' D' E' F
 Bes = Do Bes C' D' Es' F' G' A' Bes'

Dan seterusnya. Berikut adalah tabel perubahan tangga nada bermol:

Not mutlak	c	d	e	f	g	a	b	c'
Not angka	1	2	3	4	5	6	7	1
Jarak	1	1	1/2	1	1	1	1/2	
1 _b	f	g	a	bes	c'	d'	e'	f'
2 _b	bes	c'	d'	es'	f'	g'	a'	bes'
3 _b	es	f	g	as	bes	c'	d'	es'
4 _b	as	bes	c'	des'	es'	f'	g'	as'
5 _b	des	es	f	ges	as	bes	c'	des'
6 _b	ges	as	bes	ces'	des'	es'	f'	ges'
7 _b	ces	des	es	fes	ges	as	bes	ces'

Tabel 2.2 Perubahan Nada Dasar

Dalam paranada, lambang mol tidak berbeda dengan lambang kres dalam penulisannya. Lambang mol digunakan untuk menunjuk nada-nada mana saja yang diturunkan setengah laras. Lambang mol tersebut dituliskan hanya pada permulaan paranada setelah tanda kunci. Berikut adalah contoh penulisan lambang mol pada sangkar nada atau paranada:



Gambar 2.19 : Tangga Nada Mol

Sumber : (Surmani, 2004:34)

2.12 Visual Studio 2012

Basic merupakan bahasa pemrograman komputer. Bahasa pemrograman adalah perintah-perintah atau instruksi yang dimengerti oleh komputer untuk melakukan tugas-tugas tertentu. *Visual Basic* selain disebut sebagai bahasa pemrograman, juga sering disebut sebagai sarana (*tool*) untuk menghasilkan program-program aplikasi berbasis windows.

Visual Studio memungkinkan pembuatan aplikasi *Graphical User Interface* (GUI) atau pemrograman yang menggunakan tampilan grafis sebagai alat komunikasi dengan pemakainya. Dalam *Visual Studio* untuk pembuatan tampilan user interface relatif mudah dilakukan karena hanya perlu meletakkan objek-objek grafis ke sumber (*form*) yang sudah disediakan *Visual Studio*. Setelah itu cukup mengatur properti dari objek-objek tersebut. Beberapa kemampuan atau manfaat dari Visual Basic diantaranya adalah :

1. Untuk membuat program aplikasi berbasis Window
2. Untuk membuat objek-objek pembantu program seperti kontrol *ActiveX*, file *Help*, aplikasi internet dan sebagainya.
3. Menguji program (*debugging*) dan menghasilkan program akhir berakhiran *EXE* yang bersifat *Executable* atau dapat langsung dijalankan.

Terdapat *tools* pada *Visual Studio*. *Tools* tersebut diantaranya adalah :

1. Main Menu

Main menu terdiri dari dua komponen yaitu MenuBar dan TitleBar. MenuBar menampilkan menu yang berisi perintah-perintah pada *Visual Studio*, sedangkan TitleBar menampilkan judul proyek *Visual Studio* yang sedang dikerjakan.

2. Menu ToolBox

ToolBox mengandung semua objek atau kontrol yang dibutuhkan untuk membentuk suatu program aplikasi. Kontrol adalah suatu objek yang akan menjadi *interface* (penghubung) antara program aplikasi dan *user*.

3. Form Window

Form Window atau jendela form adalah daerah kerja utama, Form Window untuk meletakkan berbagai macam objek interaktif seperti teks, gambar, tombol-tombol perintah, *scrollbar* dan sebagainya. Pada saat program aplikasi dijalankan, semua yang terdapat di dalam form akan ditampilkan pada layar window. Jendela form inilah yang akan menjadi latar belakang dari program aplikasi.

4. Project Explorer

Jendela Project Explorer adalah jendela yang mengandung semua file di dalam aplikasi *Visual Studio*. Setiap aplikasi dalam *Visual Studio* disebut dengan istilah *project* (proyek), dan setiap proyek bisa mengandung lebih dari satu file. Pada Project Explorer ditampilkan semua file yang terdapat pada aplikasi (proyek), misalnya form, modul, class dan sebagainya.

5. Jendela Properties

Jendela Properties adalah jendela yang mengandung semua informasi mengenai objek yang terdapat pada aplikasi *Visual Studio*. Properti adalah sifat dari sebuah

objek, seperti nama, warna, ukuran, posisi dan sebagainya. Untuk mengatur bentuk dan karakteristik dari setiap objek melalui Jendela Properties. Jendela Properties ini memiliki dua buah tab yaitu tab *Alphabetic* dan tab *Categorized*. Pada tab *Alphabetic*, semua properti dari objek akan diurutkan berdasarkan abjad, sedangkan pada tab *Categorized*, semua properti dikelompokkan berdasarkan jenisnya. Tidak ada perbedaan antara kedua tab tersebut terhadap properti yang ditampilkan, perbedaan keduanya hanyalah cara menampilkan.

6. Form Layout Window

Form Layout Window adalah jendela yang menggambarkan posisi dari form yang ditampilkan pada layar monitor. Posisi form pada Form Layout Window inilah yang merupakan petunjuk di mana aplikasi akan ditampilkan pada layar monitor saat dijalankan. Jika ingin memindahkan form ke posisi yang lain pada layar, dengan menggeser form tersebut pada jendela Form Layout Window.

7. Jendela Code

Jendela *Code* adalah salah satu jendela yang penting dalam *Visual Studio*. Jendela ini berisi kode-kode program yang merupakan instruksi-instruksi untuk aplikasi *Visual Studio*. Setiap objek pada *Visual Studio* dapat ditambahkan dengan kode-kode program untuk melakukan tugas-tugas tertentu, misalnya menutup aplikasi, membatalkan perintah dan sebagainya. Pada saat menjalankan Visual Basic, jendela ini tidak akan ditampilkan pada layar.

2.13 Text File

Text file merupakan *file* yang berisi informasi-informasi dalam bentuk teks. Data yang berasal dari dokumen pengolah kata, angka yang digunakan dalam perhitungan, nama dan alamat dalam basis data merupakan contoh masukan data teks yang terdiri dari karakter, angka dan tanda baca. Masukan dan keluaran data teks direpresentasikan sebagai set karakter atau sistem kode yang dikenal oleh sistem komputer. Secara umum, format data teks dibagi menjadi dua bagian, yaitu :

1. Teks sederhana (plain text)

Format data teks (*.txt) merupakan contoh format teks jenis ini yang paling populer.

2. Teks terformat (formatted text) .

Merupakan teks yang terformat dan mengandung *styles*. Format data dokumen

Microsoft Word (*.doc) merupakan contoh format teks jenis ini yang paling populer.

Tipe teks merupakan tipe dasar yang sudah sangat dikenal dalam kehidupan sehari-hari. Tipe teks terdiri dari tipe karakter (*char*) dan tipe *string*. Tipe karakter (*char*) terdiri atas satu huruf, angka, tanda baca, atau karakter khusus seperti a, 1, * dan sebagainya. Tipe string terdiri atas nol atau lebih karakter seperti "algoritma", "teks" dan sebagainya.

2.14 DevExpress

Devexpress merupakan *software developent company* yang berdiri tahun 1998. Devexpress memproduksi komponen tambahan untuk borland delphi/c++ builder, activex control untuk visual studio dan html5 atau javascript.

Komponen tambahan untuk **visual studio** berisi control-control yang membantu kita untuk membuat sebuah aplikasi yang profesional (dari segi *user interface*), fitur yang lebih lengkap, dan cepat dalam mengelola data.

Devexpress memiliki komponen-komponen yang serupa dengan visual studio pada umumnya, tetapi devexpress memiliki kelebihan dari *user inteface*. Dalam penerapannya, komponen-komponen dalam devexpress menerapkan properti-properti yang berbeda dengan visual studio.

Produk yang dikeluarkan oleh devexpress diantaranya WinForm yang merupakan *UI Control* untuk windows forms, ASP.Net yang merupakan *UI Control* untuk ASP.Net dan ASP.Net MVC, WPF yang merupakan *UI Control* untuk *Windows Presentation Foundation*, *Reporting* yang merukan komponen untuk membuat *report*, *Code Rush* yang merupakan *add-ins* untuk microsoft visual studio dan yang lainnya.

BAB 3

ANALISIS DAN PERANCANGAN PERANGKAT LUNAK

3.1 Gambaran Umum Perangkat Lunak

Pada tahap ini, penulis akan memberikan gambaran umum perangkat lunak yang akan dirancang oleh penulis kepada pengguna. Secara garis besar, perangkat lunak yang akan dirancang oleh penulis membutuhkan beberapa *input data*. Pada bagian *input* ini, pengguna akan melakukan input data yang dibutuhkan, seperti : judul partitur, pengarang, nada dasar yang digunakan pengguna, birama, serta notasi-notasi angka.. Pada bagian ini pula, pengguna akan melakukan *input data* awal seperti judul, pengarang, nada dasar dan birama pada saat pengguna membuat partitur baru. Setelah proses input selesai, perangkat lunak akan menampilkan *form* baru untuk pengguna memulai proses penulisan partitur.

Pengguna melakukan input notasi dengan melalui beberapa *button* yang tersedia pada *form* tersebut. *Button* tersebut berisikan beberapa notasi yang dibutuhkan dalam pembuatan partitur seperti notasi angka (nol sampai tujuh), tanda diam, tanda kromatis (kres, mol, pugar) dan sebagainya.

Selain proses *input data* yang dilakukan pengguna, perangkat lunak ini pun akan melakukan proses konversi notasi-notasi yang telah *diinput* oleh pengguna. Proses ini akan menghasilkan gambaran notasi balok yang didasarkan pada notasi-notasi yang telah *diinput* oleh pengguna.

Dalam proses pengecekan nada-nada dari notasi yang telah ditulis oleh pengguna, perangkat lunak dapat memainkan nada-nada dari notasi yang telah ditulis oleh pengguna. Hal ini dapat membantu pengguna dalam menganalisa notasi yang telah ditulis.

Pengguna dapat menyimpan hasil partitur yang dibuatnya ke dalam sebuah *textfile*. *Textfile* ini menyimpan data notasi-notasi angka yang telah dibuat oleh pengguna dengan bentuk deretan angka dan huruf yang dipisahkan dengan tanda titik dua. Pengguna dapat membuka kembali *textfile* partitur yang telah dibuat dan pengguna dapat melakukan perubahan terhadap data partitur yang telah dibuat.

3.2 Spesifikasi Kebutuhan

Spesifikasi kebutuhan meliputi elemen-elemen atau komponen-komponen apa saja yang dibutuhkan perangkat lunak yang akan dibangun oleh penulis sampai dengan perangkat lunak tersebut diimplementasikan. Komponen-komponen tersebut dibagi menjadi dua kelompok, yaitu spesifikasi kebutuhan fungsional dan non-fungsional.

3.2.1 Analisis Fungsional

Analisis fungsional merupakan kebutuhan yang berisi penjelasan mengenai layanan-layanan atau fitur yang tersedia dalam perangkat lunak yang akan dibangun oleh penulis. Berikut adalah analisis fungsional pada perangkat lunak yang akan dibangun oleh penulis.

1. Perangkat lunak menerima masukan data-data notasi angka yaitu angka nol sampai dengan tujuh, titik dan garis sebagai tanda jumlah ketukan nada.
2. Perangkat lunak harus dapat menganalisa jumlah ketukan yang ditempatkan dalam bar sesuai dengan birama yang ditentukan.
3. Perangkat lunak harus dapat memainkan nada-nada yang ditempatkan oleh pengguna dalam mempermudah analisa nada.

3.2.2 Analisis Non-Fungsional

Pada analisis non-fungsional akan dijelaskan batasan-batasan fasilitas yang disediakan dalam menunjang perancangan perangkat lunak oleh penulis. Batasan tersebut dapat berupa spesifikasi kebutuhan perangkat lunak (*software*), spesifikasi kebutuhan perangkat keras (*hardware*) dan pengguna (*brainware*).

Perangkat lunak yang dibutuhkan oleh penulis dalam membangun dan mengoperasikan sistem adalah sebagai berikut :

1. Sistem Operasi *Microsoft Windows 7*
2. Visual Studio 2012
3. Devexpress 15.1
4. DotNet Framework 4.5

Selain perangkat lunak, spesifikasi perangkat keras yang baik juga dibutuhkan untuk menunjang pengembangan perangkat lunak ini. Spesifikasi perangkat keras yang dipakai dalam merancang perangkat lunak ini diantaranya :

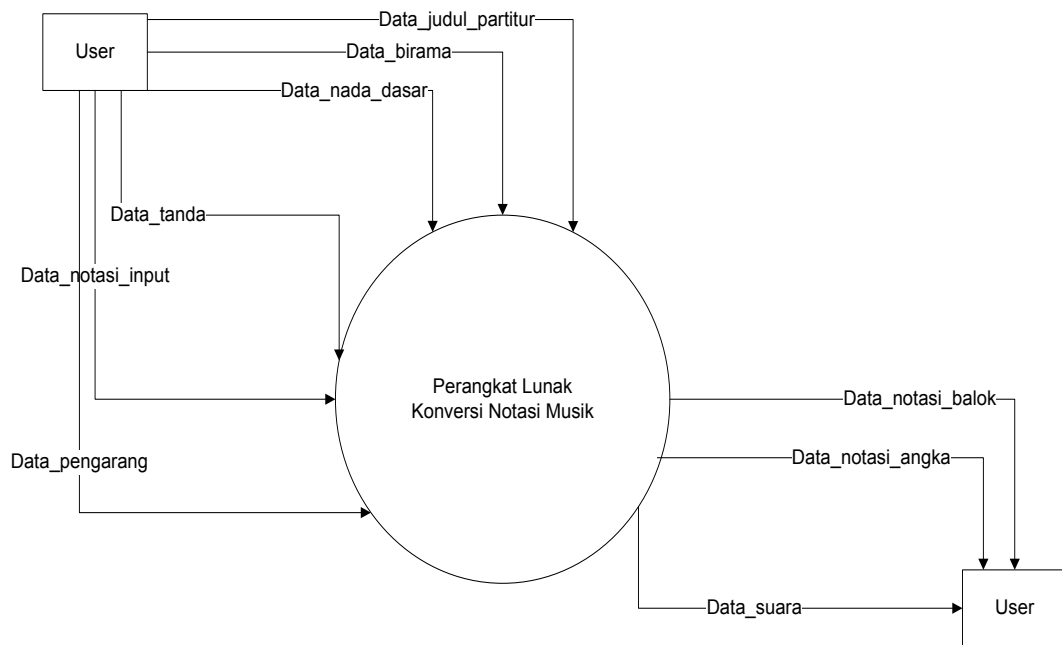
1. *Processor AMD QuadCore @2,00 Ghz*
2. RAM 1 GB
3. VGA AMD Radeon HD 8400
4. *Mouse dan keyboard*
5. Harddisk 5MB

3.3 Data Flows Diagram

Berikut adalah *data flow diagram* pada perangkat lunak konversi notasi angka ke notasi balok yang dirancang oleh penulis.

1. Diagram Konteks

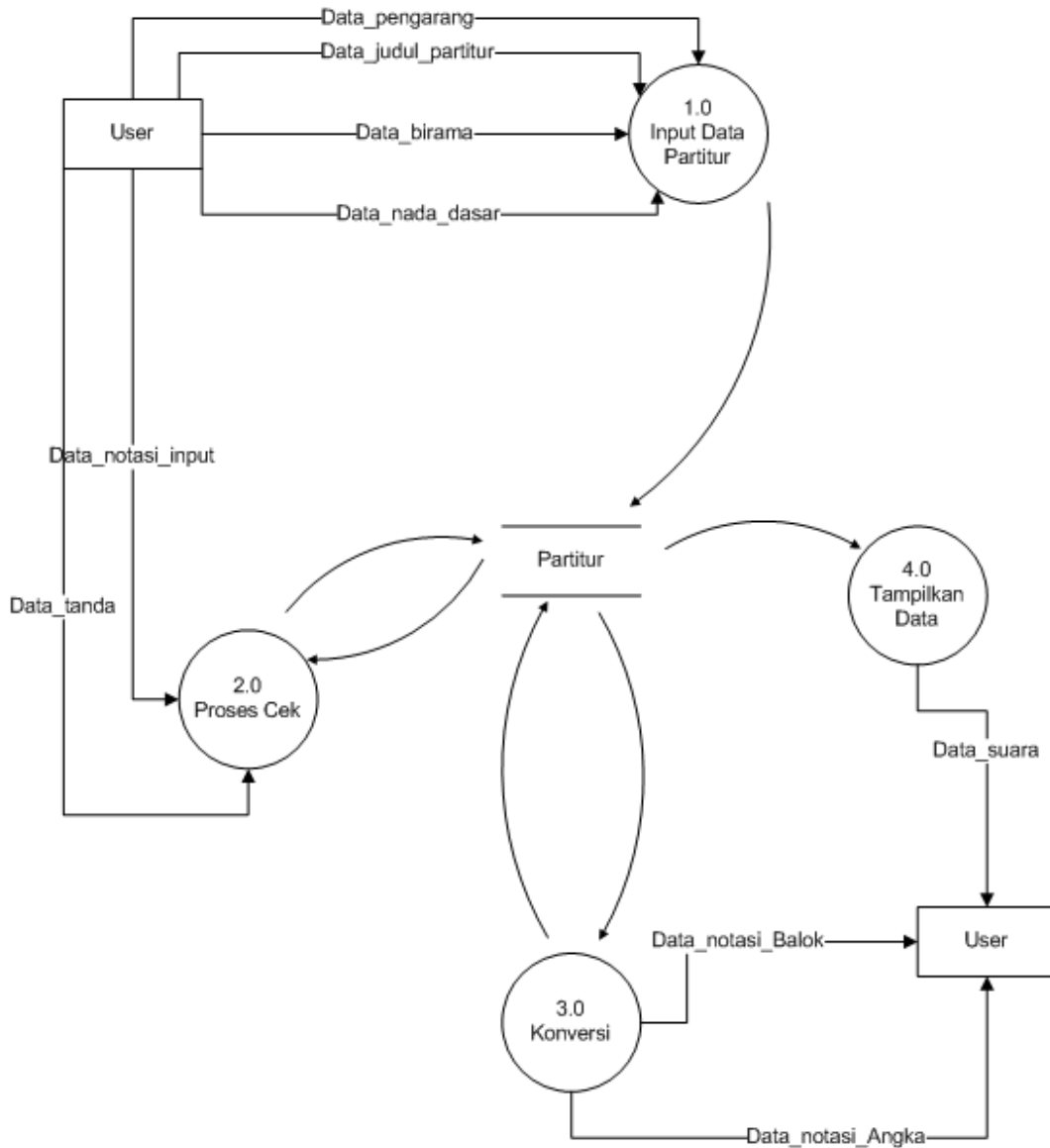
Dalam diagram alir proses konversi notasi musik dibutuhkan hanya satu entitas eksternal yaitu user. Pengguna harus mengirimkan input yaitu data-data yang berhubungan dengan partitur yaitu judul partitur, pengarang, birama, nada dasar dan notasi angka yang diinput. Perangkat lunak akan menghasilkan *output* data-data notasi angka yang telah disusun oleh pengguna, notasi balok yang terkonversi secara langsung berdasarkan data-data notasi angka yang telah disusun oleh penulis dan data suara berdasarkan data-data notasi angka.



Gambar 3.1 : Diagram Konteks

2. Diagram *Logic Level 0*

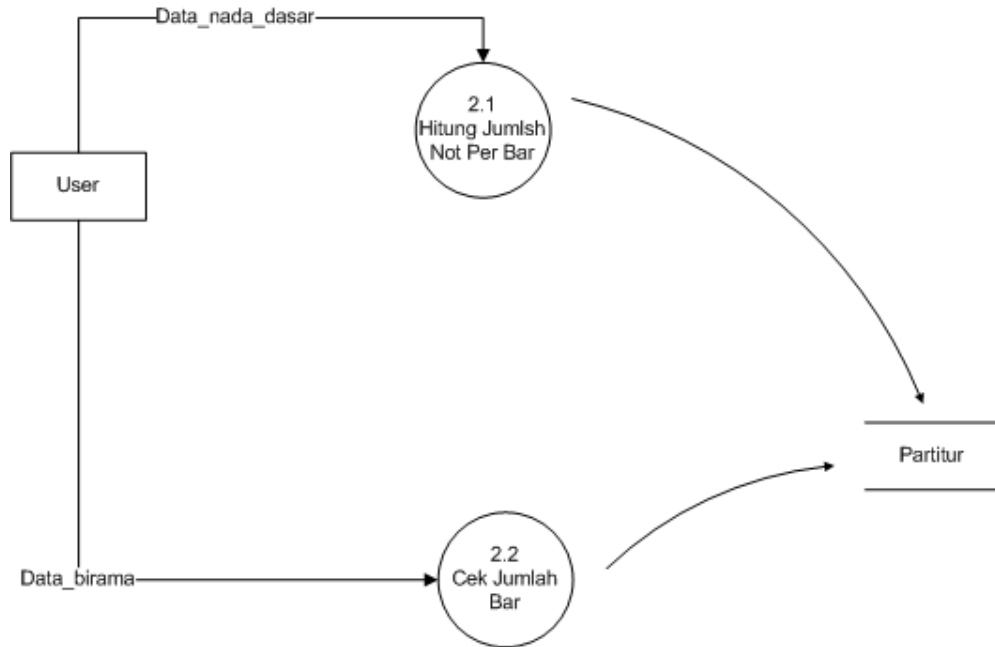
Pada diagram *logic level 0* ini, fungsi perangkat lunak dipecah menurut fungsinya masing. Pada awalnya, pengguna akan melakukan proses input data-data yang berhubungan dengan partitur, yaitu judul partitur, pengarang, nada dasar dan birama yang digunakan. Setelah selesai, pengguna akan melakukan *input* notasi-notasi angka dan perangkat lunak akan melakukan proses cek ketukan notasi-notasi angka dalam satu bar dan jumlah bar. Ketukan dalam satu bar harus sesuai dengan birama yang ditentukan sejak awal oleh pengguna. Perangkat lunak pun melakukan inisialisasi posisi notasi balok yang akan di gambarkan dan juga perangkat lunak dapat melakukan proses analisa nada-nada yang ditempatkan oleh pengguna agar pengguna dapat mengetahui nada-nada yang ditulis. Setelah selesai input data dan proses cek, maka perangkat lunak melakukan proses konversi dari data yang dibuat ke dalam bentuk notasi balok, notasi angka dan juga suara berdasarkan notasi yang sudah dibuat oleh pengguna. Penjelasan dalam diagram alir adalah sebagai berikut.



Gambar 3.2 : Diagram *Logic Level 0*

3. Diagram *Logic Level 1* (Proses cek)

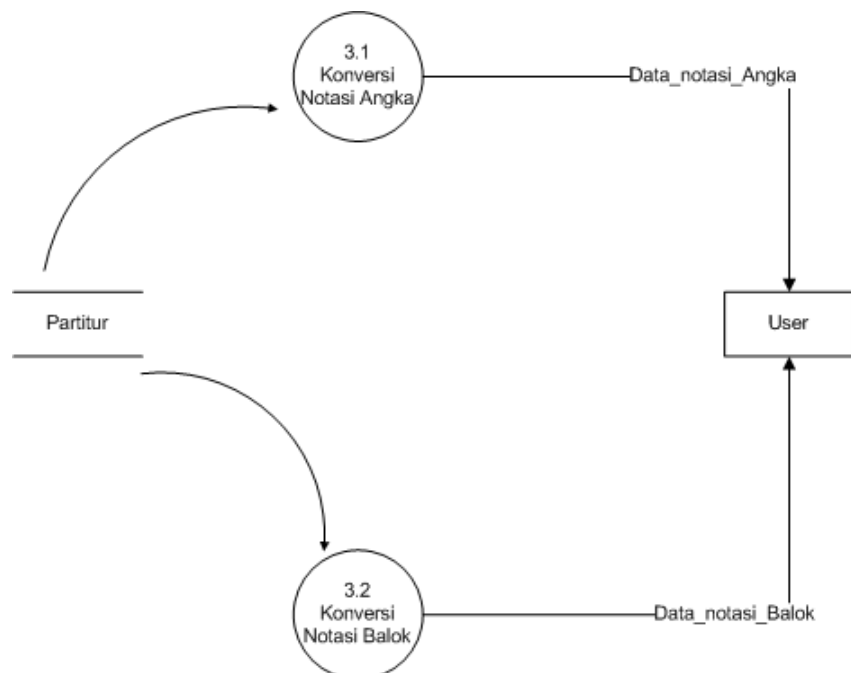
Pada tahap ini, proses akan dipecah ke dalam beberapa subproses. Input data-data notasi akan dilakukan cek, apakah banyaknya ketukan dalam satu bar sesuai dengan birama yang ditentukan atau tidak dan apakah jumlah bar dalam satu garis paranada sesuai dengan birama yang telah di *input* oleh pengguna. Penjelasan dalam diagram alir adalah sebagai berikut.



Gambar 3.3 : Diagram *Logic Proses Cek Level 1*

4. Diagram *Logic Level 1* (Proses Konveersi)

Pada tahap ini, proses akan dipecah ke dalam beberapa subproses. Perangkat lunak akan mengambil data-data dari *file* dan melakukan proses konversi notasi angka maupun konversi notasi balok. Penjelasan dalam diagram alir adalah sebagai berikut.



Gambar 3.4 : Diagram *Logic Proses Konversi Level 1*

3.4 Spesifikasi Proses

Process Specification atau disingkat PSPEC digunakan untuk menggambarkan algoritma dari semua proses model yang menjadi level terendah. PSPEC dari perangkat lunak ini adalah :

1. Proses *Input* Data Partitur

Fungsi : Melakukan *input* data partitur awal

Input : Data judul partitur, pengarang, nada dasar dan birama

Output : Data *input* menjadi acuan dalam proses selanjutnya

Tipe Proses : Online

Logika Proses :

Begin

 Input Data;

 If Judul or Pengarang or Nada Dasar or Birama = null then

 Muncul Dialog Pesan Peringatan;

 Else

 Simpan Data;

 Draw Data;

 Show Input Notasi Form;

End.

2. Proses Cek

2.1 Proses Hitung Jumlah Not per Bar

Fungsi : Untuk melakukan analisa ketukan dalam satu bar

Input : Data birama, Data notasi input

Output : Data Separator Bar

Tipe Proses : Online

Logika Proses :

Begin

 GetDataBirama();

```

GetDataNotasiInput();
IF Jumlah DataNotasiInput < DataBirama Then
    Hitung Jumlah Notasi Input;
ELSE Jumlah DataNotasiInput >= DataBirama Then
    Draw Garis Birama();
    Insert Separator Bar To File();
ENDIF
End

```

2.2 Proses Cek Jumlah Bar

Fungsi : Untuk melakukan analisa jumlah bar dalam satu garis paranada

Input : Data birama, Data notasi input

Output : Data Separator Birama

Tipe Proses : Online

Logika Proses :

```

Begin
    GetDataBirama();
    GetDataNotasiInput();
    IF Jumlah Data Separator Bar < DataBirama Then
        Hitung Jumlah Bar;
    ELSE Jumlah Data Separator Bar >= DataBirama Then
        Draw New Paranada();
        Insert Separator Birama To File();
    ENDIF
End

```

3. Proses Konversi Notasi

3.1 Proses Konver Notasi Angka

Fungsi : Melakukan konversi notasi angka

Input : Data notasi

Output : Data notasi angka

Tipe Proses : Online

Logika Proses :

Begin

 GetDataInput();

 While not end of notasi

 Konversi Notasi Angka();

 Show Partitur Angka();

 Endwhile

End

3.2 Proses Konver Notasi Balok

Fungsi : Melakukan konversi notasi Balok

Input : Data notasi, tanda, birama, dan tangga nada

Output : Data notasi balok

Tipe Proses : Online

Logika Proses :

Begin

 GetDataInput();

 While not end of Data Notasi

 Konversi Notasi Balok();

 Show Partitur Balok();

 Endwhile

End

4. Proses Tampilkan Data

Fungsi : Menampilkan data suara

Input : Data partitur

Output : Data suara

Tipe Proses :

Logika Proses :

Begin

 Get Data Partitur();

 While not end of notasi

 Play Data Suara();

 End While;

End

3.5 Struktur Data

Pada tahap ini, penulis akan memberikan gambaran umum struktur data perangkat lunak yang dirancang oleh penulis. Secara garis besar, struktur data perangkat lunak yang dirancang oleh penulis berupa sekumpulan huruf dan angka yang saling berhubungan satu sama lain dan dipisahkan dengan titik dua.

Pada baris pertama, data memiliki arti data judul, pengarang, birama dan nada dasar yang di pisahkan dengan tanda titik dua. Sebagai contoh :

Elshadai : Priskila : 4/4 : D:

Pada baris berikutnya adalah serangkaian data untuk mendeklarasikan notasi dan tanda. Berikut adalah penjelasannya

1. Jenis Notasi

W = Notasi dengan empat ketuk.

H = Notasi dengan dua ketuk.

Q = Notasi dengan satu ketuk.

E = Notasi dengan setengah ketuk.

S = Notasi dengan seperempat ketuk.

2. *Range* Notasi

T = Notasi dengan *range* tinggi.

N = Notasi dengan *range* normal.

R = Notasi dengan *range* rendah.

3. Nada

1 = Menunjukkan nada do.

2 = Menunjukkan nada re.

3 = Menunjukkan nada mi.

4 = Menunjukkan nada fa.

5 = Menunjukkan nada sol.

6 = Menunjukkan nada la.

7 = Menunjukkan nada si.

4. Tanda

: = Menunjukkan pemisah setiap data.

C = Menunjukkan tanda kres.

N = Menunjukkan notasi tidak memiliki penanda.

B = Menunjukkan akhir bar.

END = Menunjukkan akhir paranada.

ENDPART = Menunjukkan akhir partitur.

Data notasi tersebut akan dituliskan dalam bentuk text yang berurutan. Sebagai contoh penulisan data partitur adalah sebagai berikut :

W N 5 N : B : END : ENDPART :

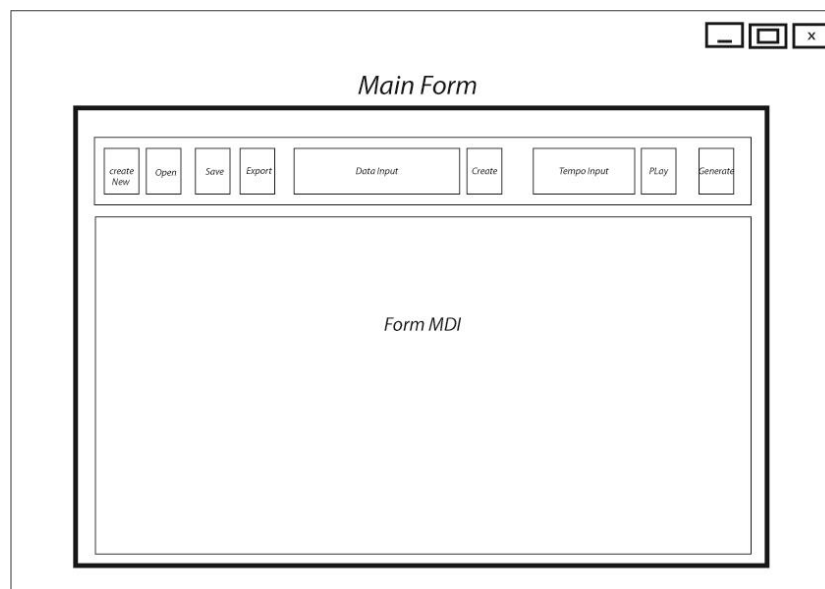
Huruf *W* menunjukkan jenis notasi yang berupa *whole note*, huruf *N* menunjukkan bahwa angka lima adalah nada sol pada *range* normal. Huruf *N* selanjutnya menandakan notasi sol tidak memiliki tanda. Titik dua adalah pemisah antara notasi. *B* menunjukkan akhir birama, *END* menunjukkan akhir garis paranada dan *ENDPART* menunjukkan akhir dari file partitur tersebut.

3.6 Tampilan Antar Muka

Berikut adalah rancangan program antar muka dari aplikasi konversi notasi musik yang akan dibuat :

1. Tampilan antar muka Main Form

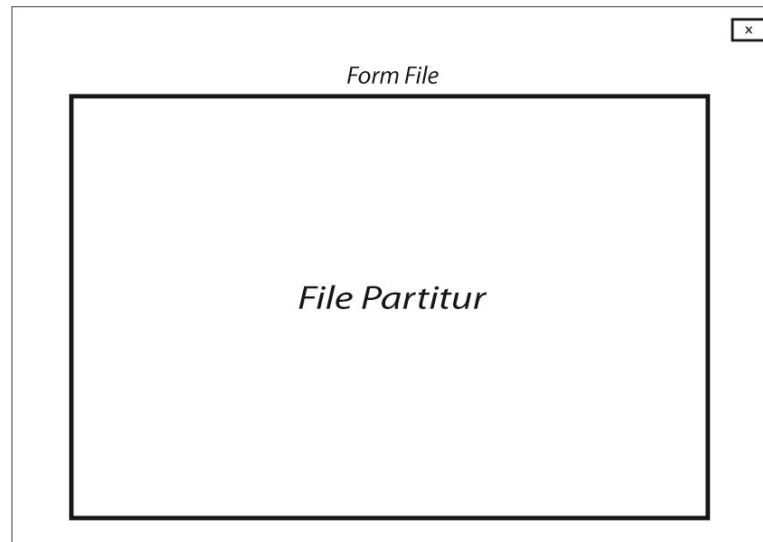
Pada *main form*, tombol-tombol, *combobox* dan *textedit* dikelompokkan pada *ribbonbar* yang berada pada bagian atas *main form*. Tombol-tombol tersebut berfungsi untuk membuat partitur baru, mengambil data partitur yang telah tersimpan, melakukan penyimpanan data partitur yang telah pengguna buat, melakukan *export* partitur notasi angka dan partitur notasi balok ke dalam bentuk gambar, memainkan nada-nada dari data notasi yang telah dibuat pengguna dan tombol untuk melakukan *generate* partitur notasi angka dan partitur notasi balok dari data notasi yang pengguna sudah *input*. *Textedit* pada *main form* ini berfungsi sebagai data masukan untuk membuat partitur baru yang diantaranya data judul partitur dan data pengarang partitur. *Combobox* pada *mainform* berfungsi sebagai masukan untuk data birama partitur, data nada dasar partitur dan data masukan tempo sebagai acuan waktu dalam memainkan notasi yang telah ditulis pengguna.



Gambar 3.5 : Tampilan Antar Muka *Main Form*

2. Tampilan antar muka Form File

Pada *form* ini, terdapat satu *textedit* ataupun *richedit* yang berfungsi untuk menampilkan data notasi yang telah ditulis oleh pengguna. Data-data yang ditampilkan pada dialog tersebut akan menjadi acuan dalam proses *generate* notasi ke dalam bentuk partitur notasi balok dan notasi angka.



Gambar 3.6 : Tampilan Antar Muka *Form File*

3. Tampilan antar muka Form Notasi Angka

Pada *form* notasi angka, terdapat tombol-tombol yang fungsinya terbagi menjadi tiga, yaitu sebagai masukan jumlah ketukan tiap notasi yang *diinput* oleh pengguna, sebagai masukan notasi dan juga sebagai masukan tanda-tanda notasi. Selain itu, terdapat tombol yang berfungsi sebagai pemberi tanda akhir, dan akhir tersebut menjadi acuan bahwa partitur sudah selesai dibuat oleh pengguna. Terdapat sebuah dialog juga untuk menampilkan notasi yang telah diinput oleh pengguna.

Gambar 3.7 : Tampilan Antar Muka *Form Notasi Angka*

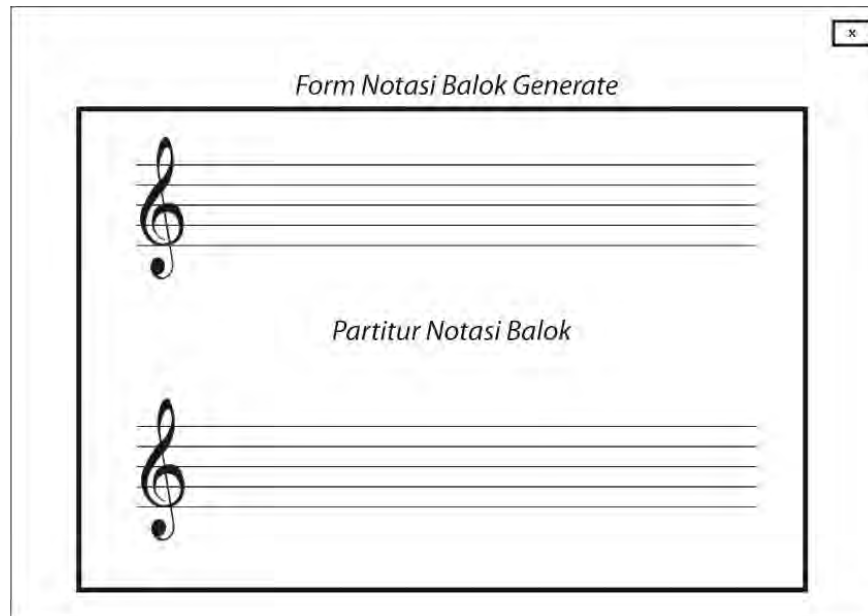
4. Tampilan antar muka Form Notasi Angka Generate

Pada form ini, hanya terdapat satu buah dialog untuk menampilkan partitur notasi angka yang telah melalui proses *generate* dengan acuan data-data yang telah *diinput* oleh pengguna.

Gambar 3.8 : Tampilan Antar Muka *Form Notasi Angka Generate*

5. Tampilan antar muka Form Notasi Balok Generate

Pada form ini, hanya terdapat satu buah dialog untuk menampilkan partitur notasi balok yang telah melalui proses *generate*.



Gambar 3.9 : Tampilan Antar Muka *Form Notasi Balok Generate*

BAB 4

IMPLEMENTASI DAN PENGUJIAN PERANGKAT LUNAK

4.1 Spesifikasi Kebutuhan Perangkat Keras

Pembuatan program ini menggunakan sebuah laptop dengan spesifikasi sebagai berikut :

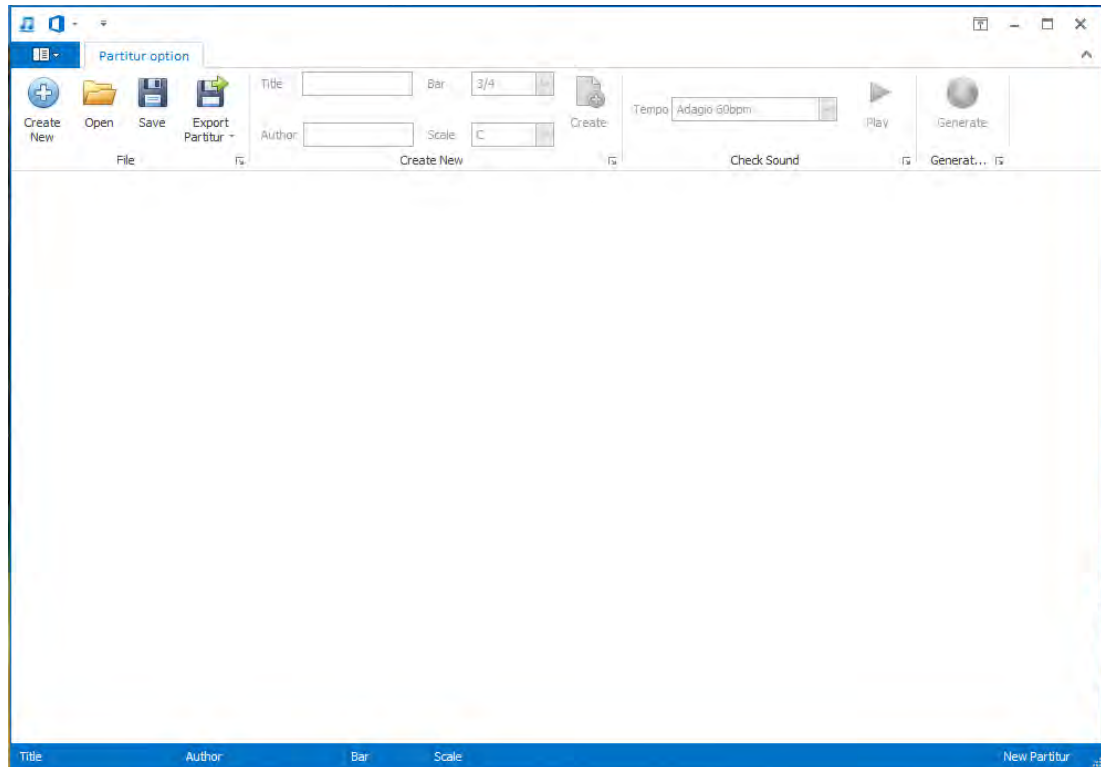
1. Processor : AMD Quad-Core Processor A6-5200 2.0 GHz
2. VGA Card : AMD Radeon™ HD 8400
3. Memori : 1GB – DDR3
4. Hard disk : 5MB
5. Monitor : 14" HD dengan resolusi 1366x768
6. Sistem operasi : Windows 7 Ultimate 64-bit
7. Dilengkapi dengan Visual Basic 2012
8. Dilengkapi dengan Devexpress 15.1
9. Dilengkapi dengan DotNet Framework 4.5

4.2 Tampilan Antar Muka

Berikut adalah *screen shoot* seluruh kemungkinan yang terjadi ketika aplikasi ini sedang berjalan :

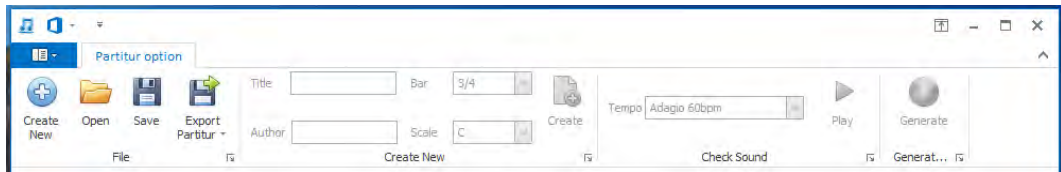
Main Form

Pada *main form*, pengguna dapat memilih berbagai menu yang terdapat pada *toolbar form* ini. Pengguna dapat membuat partitur baru maupun mengambil data partitur yang sudah pernah dibuat oleh pengguna. Setelah selesai, pengguna dapat menyimpan data partitur tersebut dan juga meng*export* data partitur notasi angka maupun notasi balok ke dalam bentuk gambar sesuai kebutuhan pengguna.



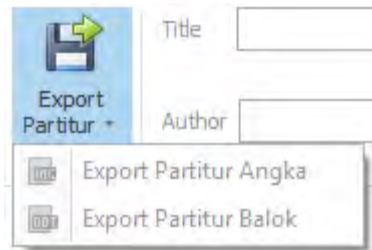
Gambar 4.1 : Main Form

Pada *group create new* yang terdapat pada *toolbar main form* terdapat *button*, dua *textedit* dan dua *combobox* sebagai masukan data partitur saat pengguna membuat data partitur baru. *Button* pada *group* ini berfungsi untuk memanggil form notasi angka setelah semua data partitur terisi. *Textedit* tersebut merupakan acuan sebagai masukan data judul partitur dan pengarang partitur. *Combobox* berfungsi sebagai acuan masukan data birama partitur dan nada dasar partitur. Pada *group check sound* terdapat satu *combobox* dan *button*. *Combobox* tersebut berfungsi sebagai data masukan tempo dan *button play* berfungsi untuk memainkan nada-nada yang telah dibuat oleh pengguna. *Button generate* berfungsi untuk melakukan *generate* data yang telah dibuat oleh pengguna ke dalam bentuk partitur notasi angka maupun partitur notasi balok.



Gambar 4.2 : *Toolbar Main Form*

Pada *button export*, saat *button* diklik, maka akan muncul *popup menu* yang berisi dua buah *button* yaitu *button* untuk melakukan *export* partitur notasi angka dan partitur notasi balok. *Button* tersebut akan dapat dipakai bila pengguna sudah melakukan proses *generate* notasi.



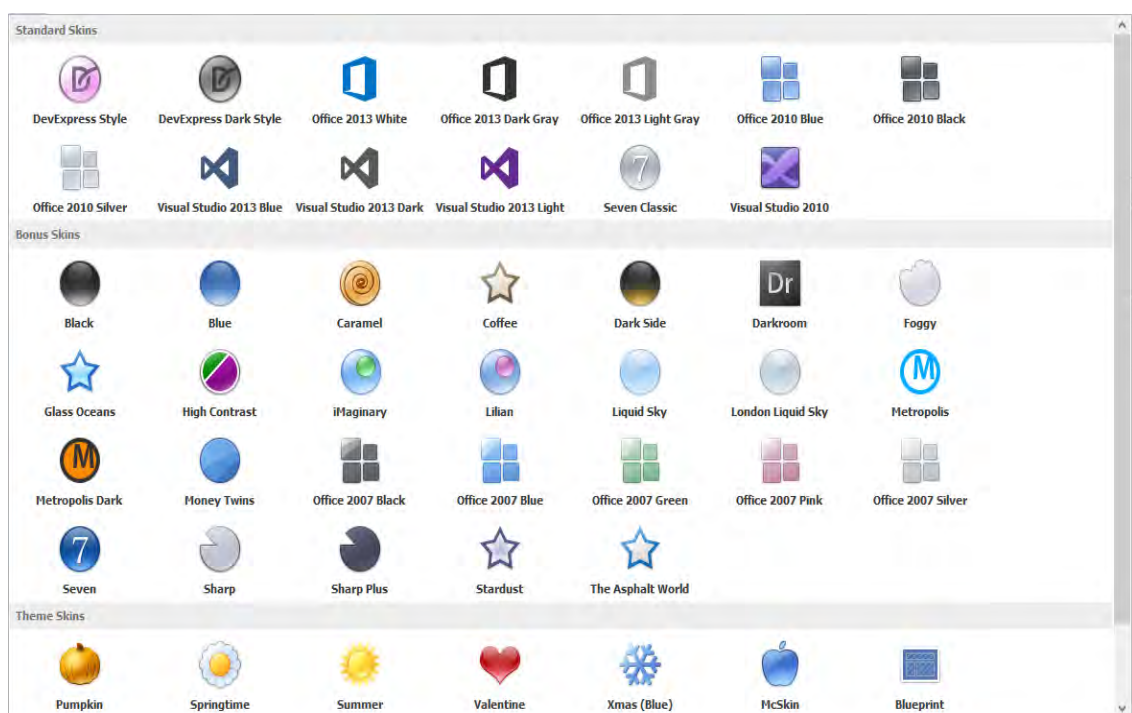
Gambar 4.3 : *Popup Menu Export Button*

Radial menu akan muncul saat pengguna melakukan klik pada *button generate* maupun pada *form file* saat partitur sudah selesai dibuat oleh pengguna. *Radial menu* ini berisikan dua buah *button* yaitu *button* untuk melakukan proses *generate* partitur notasi angka dan partitur notasi balok.



Gambar 4.4 : Menu Generate Button

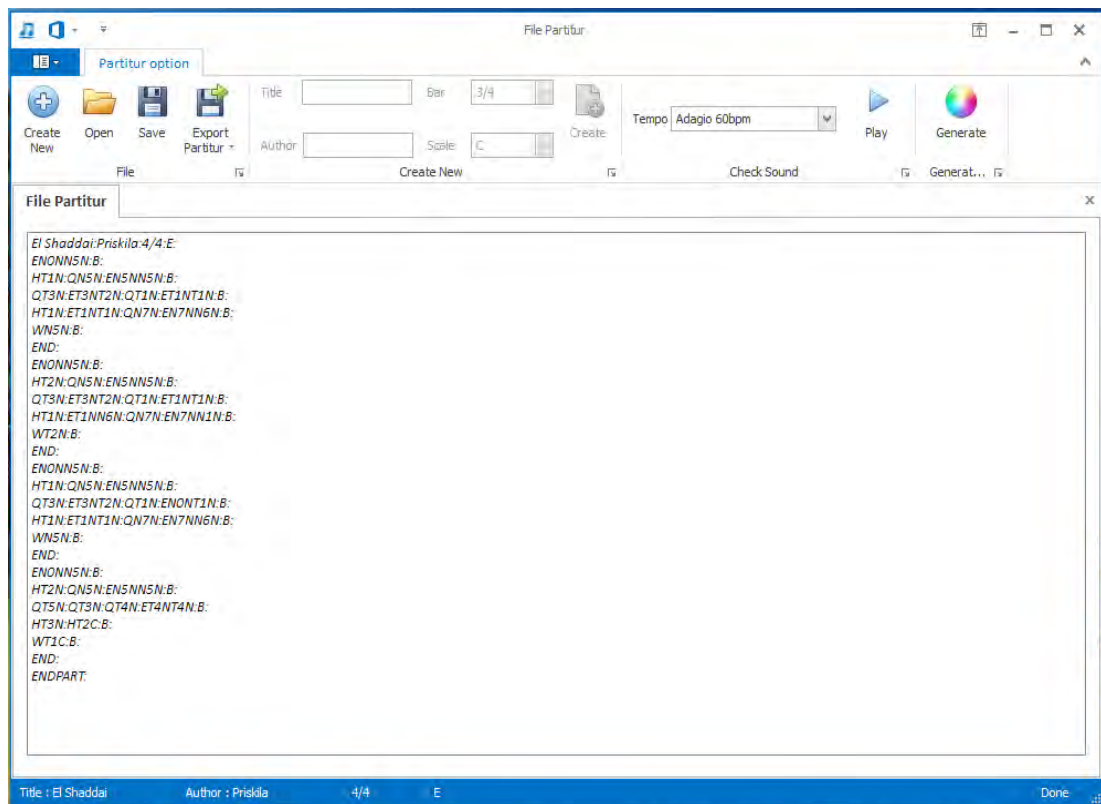
Pada *main form* juga, terdapat *button theme gallery* yang berisikan macam-macam *theme* untuk merubah tampilan *form* sesuai keinginan pengguna. Pada *gallery* ini, tersedia berbagai macam *theme* yang dapat digunakan sesuai keinginan pengguna.



Gambar 4.5 : Theme Gallery

Form File

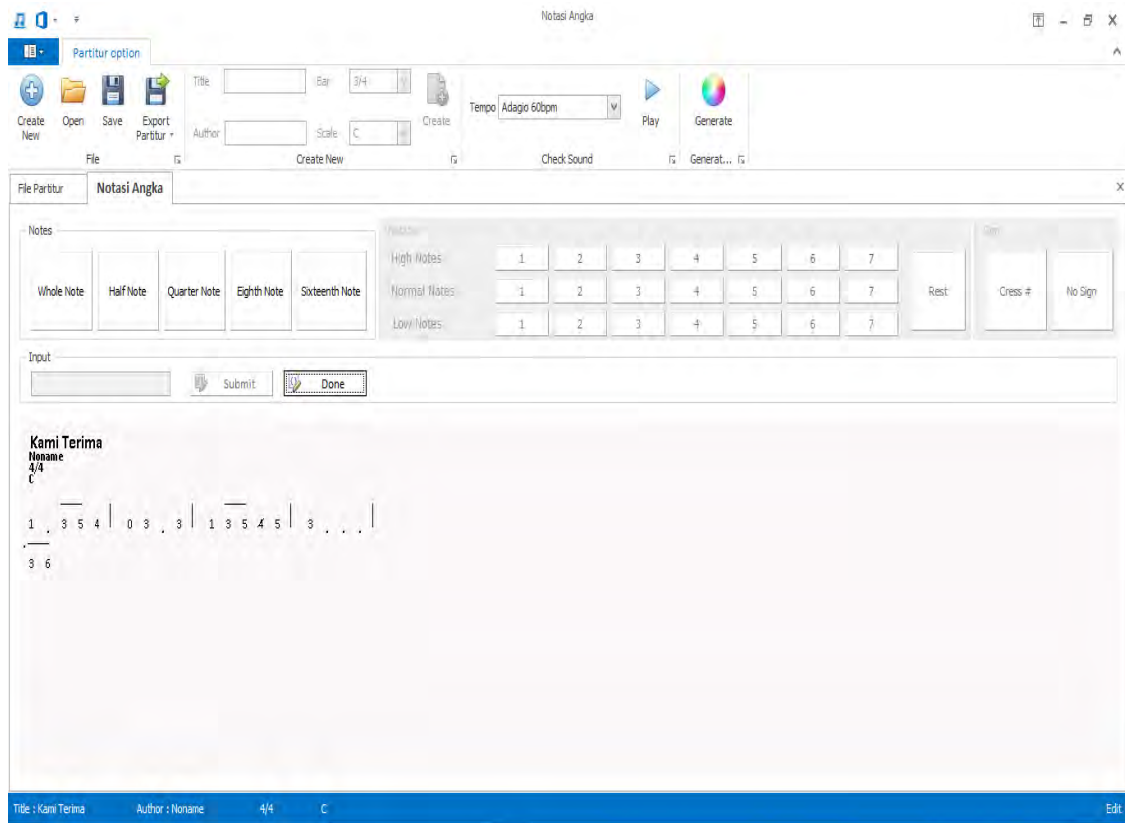
Pada bagian ini, *form* berisikan sebuah *listbox* yang berfungsi sebagai media penyimpanan sementara dari data notasi yang sudah dibuat oleh pengguna maupun yang telah pengguna ambil dari data yang sudah dibuat sebelumnya.



Gambar 4.6 : Form File

Form Notasi Angka

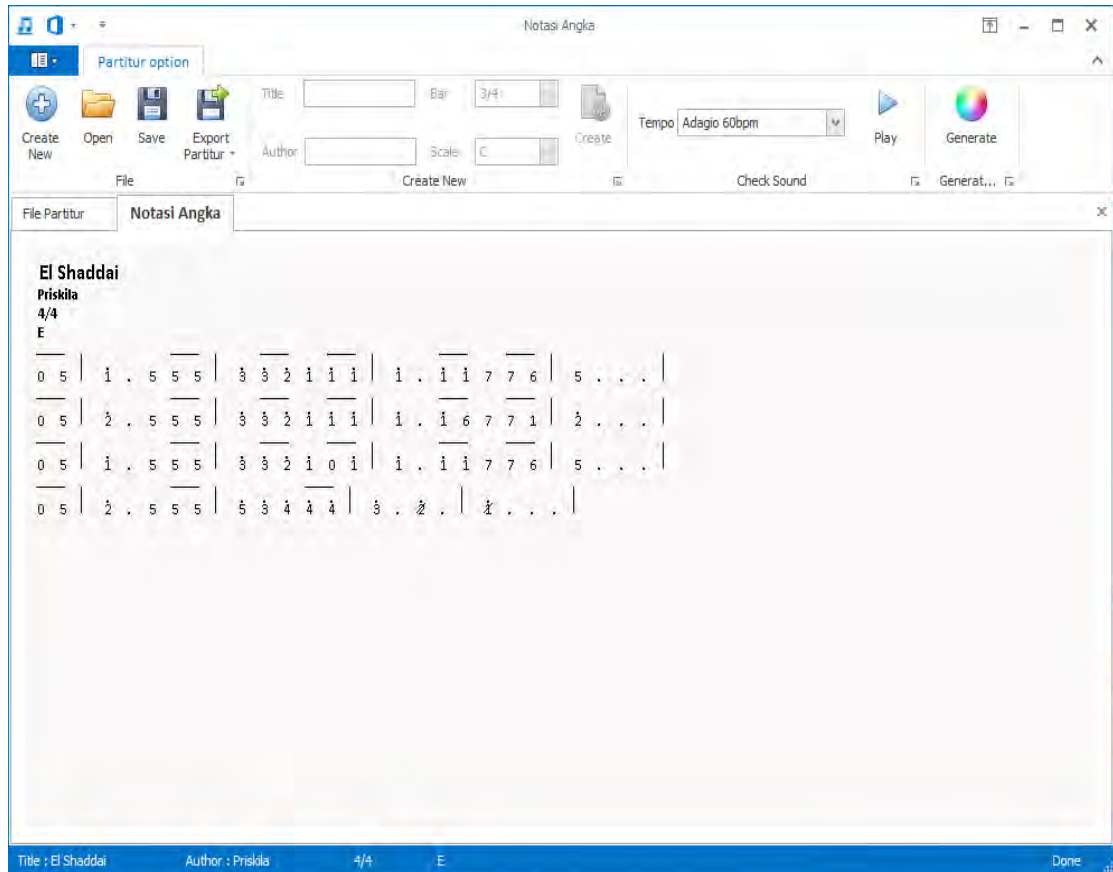
Pada form notasi angka, pengguna dapat melakukan proses *input* data notasi yang dibutuhkan. Terdapat tiga *group button* yaitu *group notes* yang berisikan *button* untuk input jumlah ketukan setiap nada yang dibutuhkan, *group notation* yang berisikan *button* untuk input data-data notasi dengan berbagai macam *range* nada dan tanda diam dan *group sign* yang berisikan *button* yang berfungsi sebagai masukan data tanda notasi. Pengguna harus melakukan proses *submit* saat notasi sudah dibuat dan menekan *button done* bila pengguna sudah selesai membuat partitur yang diinginkan untuk dapat melakukan proses *generate* partitur.



Gambar 4.7 : *Form* Notasi Angka

Form Notasi Angka Generate

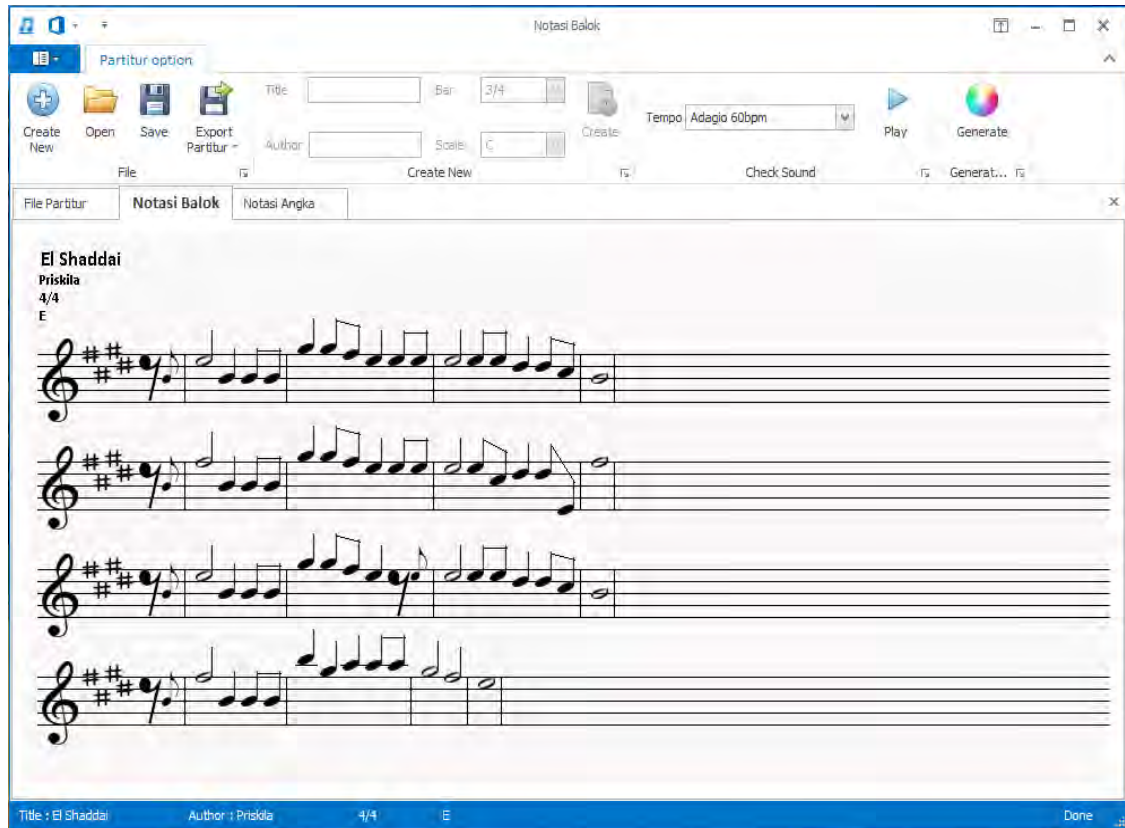
Pada *form* ini, pengguna dapat melihat partitur notasi angka yang telah dibuat setelah pengguna melakukan proses *generate*. Pengguna dapat melakukan *export* data partitur angka kedalam bentuk gambar bila *form* ini telah ditampilkan.



Gambar 4.8 : *Form Notasi Angka Generate*

Form Notasi Balok Generate

Pengguna dapat melihat partitur notasi balok yang telah dibuat setelah pengguna melakukan proses *generate*. Pengguna dapat melakukan *export* data partitur balok kedalam bentuk gambar bila *form* ini telah ditampilkan. *Form* ini berisikan satu buah *pictureedit* yang digunakan sebagai media penempatan notasi balok.



Gambar 4.9 : *Form Notasi Balok Generate*

4.3 Pengujian Fungsi

Pengujian perangkat lunak dilakukan dengan menggunakan metode black box, sehingga pengujian fungsi-fungsi perangkat lunak menjadi fokus utama. Pengujian dilakukan dengan maksud untuk melihat apakah fungsi-fungsi perangkat lunak sudah berjalan dengan sebagaimana mestinya atau belum. Berikut ini adalah tabel-tabel dari fungsi perangkat lunak yang diuji :

1. Pengujian tombol-tombol dan *main form*

No	Aksi	Hasil yang diharapkan	Output	Ket
1	Klik tombol <i>create new</i>	Mengaktifkan <i>ribbon page group create new</i>	<i>Ribbon page group create new</i> aktif	OK
2	Klik tombol <i>open</i>	Membuka dialog <i>open file</i>	Dialog <i>open file</i> terbuka	OK
3	Klik tombol <i>open</i> pada dialog	Membuka form file dan menambahkan data pada listbox	form file terbuka dan data pada listbox ditambahkan	OK
4	Klik tombol <i>cancel</i> pada dialog	Menutup dialog <i>open</i>	Dialog tertutup <i>open</i>	OK
5	Klik tombol <i>save</i>	Membuka dialog <i>save file</i>	Dialog <i>save file</i> terbuka	OK
6	Klik tombol <i>save</i> pada dialog	Menyimpan <i>file text</i>	<i>File text</i> tersimpan	OK
7	Klik tombol <i>cancel</i> pada dialog	Menutup dialog <i>save</i>	Dialog tertutup <i>save</i>	OK
8	Klik tombol <i>export</i>	Menampilkan <i>popup menu export</i>	<i>Popup menu export</i> terbuka	OK
9	Klik tombol <i>export</i> notasi angka	Menampilkan dialog <i>export partitur angka</i>	Dialog <i>export partitur angka</i> terbuka	OK
10	Klik tombol <i>save</i> pada dialog	Menyimpan <i>file jpeg notasi angka</i>	<i>File jpeg notasi angka</i> tersimpan	OK
11	Klik tombol <i>cancel</i> pada dialog	Menutup dialog <i>save</i>	Dialog tertutup <i>save</i>	OK
12	Klik tombol <i>export</i> notasi balok	Menampilkan dialog <i>export partitur balok</i>	Dialog <i>export partitur balok</i> terbuka	OK
13	Klik tombol <i>save</i> pada dialog	Menyimpan <i>file jpeg notasi balok</i>	<i>File jpeg notasi balok</i> tersimpan	OK
14	Klik tombol <i>cancel</i> pada dialog	Menutup dialog <i>save</i>	Dialog tertutup <i>save</i>	OK
15	Klik tombol <i>create</i> saat <i>textedit</i> judul atau pengarang kosong	Menampilkan pesan peringatan	Pesan peringatan muncul	OK
16	Klik tombol <i>create</i> saat <i>textedit</i> judul atau pengarang terisi	Menampilkan <i>form file</i> dan notasi angka	<i>Form file</i> dan notasi angka muncul	OK
17	Klik tombol <i>play</i>	Memainkan suara dari nada-nada	Suara dari nada-nada dimainkan	OK
18	Klik tombol <i>generate</i>	Menampilkan <i>radial menu generate</i>	<i>Radial menu generate</i> muncul	OK
19	Klik tombol <i>generate</i> notasi angka	Menampilkan <i>form not angka generate</i>	<i>Form not angka not angka</i> tampil	OK
20	Klik tombol <i>generate</i> notasi balok	Menampilkan <i>form not balok</i>	<i>Form not balok not angka</i> tampil	OK
21	Klik tombol <i>close</i>	Tutup program	Tutup program	OK
22	<i>Main form load</i>	Tombol <i>export</i> dan <i>ribbon page group create disable</i>	Tombol <i>export</i> dan <i>ribbon page group create disable</i>	OK
23	Klik tombol <i>theme</i>	Muncul dialog <i>theme gallery</i>	Dialog <i>theme gallery</i> muncul	OK

Tabel 4.1 Tabel pengujian *tombol-tombol* dan *main form*

2. Pengujian *form file*

No	Aksi	Hasil yang diharapkan	Output	Ket
1	<i>Form show</i>	<i>Text</i> judul, pengarang, bar, scale dan status <i>status bar</i> terganti	<i>Text</i> judul, pengarang, bar, scale dan status <i>status bar</i> terganti	OK
2	<i>Form close</i>	Semua <i>form</i> tertutup	Semua <i>form</i> tertutup	OK
3	Klik <i>listbox</i>	Muncul <i>radial menu generate</i>	<i>Radial menu generate</i> muncul	OK

Tabel 4.2 Tabel pengujian *form file*3. Pengujian tombol-tombol dan *form* notasi angka

No	Aksi	Hasil yang diharapkan	Output	Ket
1	Klik tombol <i>whole not</i>	Menambahkan <i>input W</i> pada <i>textbox</i>	Muncul huruf W pada <i>textbox</i>	OK
2	Klik tombol <i>half not</i>	Menambahkan <i>input H</i> pada <i>textbox</i>	Muncul huruf H pada <i>textbox</i>	OK
3	Klik tombol <i>quater not</i>	Menambahkan <i>input Q</i> pada <i>textbox</i>	Muncul huruf Q pada <i>textbox</i>	OK
4	Klik tombol <i>eight not</i>	Menambahkan <i>input E</i> pada <i>textbox</i>	Muncul huruf E pada <i>textbox</i>	OK
5	Klik tombol <i>sixteenth not</i>	Menambahkan <i>input S</i> pada <i>textbox</i>	Muncul huruf S pada <i>textbox</i>	OK
6	Klik tombol H1	Menambahkan <i>input H1</i> pada <i>textbox</i>	Huruf H1 pada <i>textbox</i> ditambahkan	OK
7	Klik tombol H2	Menambahkan <i>input H2</i> pada <i>textbox</i>	Huruf H2 pada <i>textbox</i> ditambahkan	OK
8	Klik tombol H3	Menambahkan <i>input H3</i> pada <i>textbox</i>	Huruf H3 pada <i>textbox</i> ditambahkan	OK
9	Klik tombol H4	Menambahkan <i>input H4</i> pada <i>textbox</i>	Huruf H4 pada <i>textbox</i> ditambahkan	OK
10	Klik tombol H5	Menambahkan <i>input H5</i> pada <i>textbox</i>	Huruf H5 pada <i>textbox</i> ditambahkan	OK
11	Klik tombol H6	Menambahkan <i>input H6</i> pada <i>textbox</i>	Huruf H6 pada <i>textbox</i> ditambahkan	OK
12	Klik tombol H7	Menambahkan <i>input H7</i> pada <i>textbox</i>	Huruf H7 pada <i>textbox</i> ditambahkan	OK
13	Klik tombol N1	Menambahkan <i>input N1</i> pada <i>textbox</i>	Huruf N1 pada <i>textbox</i> ditambahkan	OK
14	Klik tombol N2	Menambahkan <i>input N2</i> pada <i>textbox</i>	Huruf N2 pada <i>textbox</i> ditambahkan	OK
15	Klik tombol N3	Menambahkan <i>input N3</i> pada <i>textbox</i>	Huruf N3 pada <i>textbox</i> ditambahkan	OK
16	Klik tombol N4	Menambahkan <i>input N4</i> pada <i>textbox</i>	Huruf N4 pada <i>textbox</i> ditambahkan	OK
17	Klik tombol N5	Menambahkan <i>input N5</i> pada <i>textbox</i>	Huruf N5 pada <i>textbox</i> ditambahkan	OK
18	Klik tombol N6	Menambahkan <i>input N6</i> pada <i>textbox</i>	Huruf N6 pada <i>textbox</i> ditambahkan	OK

19	Klik tombol N7	Menambahkan <i>input</i> N7 pada <i>textbox</i>	Huruf N7 pada <i>textbox</i> ditambahkan	OK
20	Klik tombol R1	Menambahkan <i>input</i> R1 pada <i>textbox</i>	Huruf R1 pada <i>textbox</i> ditambahkan	OK
21	Klik tombol R2	Menambahkan <i>input</i> R2 pada <i>textbox</i>	Huruf R2 pada <i>textbox</i> ditambahkan	OK
22	Klik tombol R3	Menambahkan <i>input</i> R3 pada <i>textbox</i>	Huruf R3 pada <i>textbox</i> ditambahkan	OK
23	Klik tombol R4	Menambahkan <i>input</i> R4 pada <i>textbox</i>	Huruf R4 pada <i>textbox</i> ditambahkan	OK
24	Klik tombol R5	Menambahkan <i>input</i> R5 pada <i>textbox</i>	Huruf R5 pada <i>textbox</i> ditambahkan	OK
25	Klik tombol R6	Menambahkan <i>input</i> R6 pada <i>textbox</i>	Huruf R6 pada <i>textbox</i> ditambahkan	OK
26	Klik tombol R7	Menambahkan <i>input</i> R7 pada <i>textbox</i>	Huruf R7 pada <i>textbox</i> ditambahkan	OK
27	Klik tombol <i>rest</i>	Menambahkan <i>input</i> N0 pada <i>textbox</i>	Huruf N0 pada <i>textbox</i> ditambahkan	OK
28	Klik tombol <i>cross</i>	Menambahkan <i>input</i> C pada <i>textbox</i>	Huruf C pada <i>textbox</i> ditambahkan	OK
29	Klik tombol <i>no sign</i>	Menambahkan <i>input</i> N pada <i>textbox</i>	Huruf N pada <i>textbox</i> ditambahkan	OK
30	Klik tombol <i>submit</i>	<i>Input</i> notasi angka pada <i>picture edit</i>	Notasi angka terinput	OK
31	Klik tombol <i>done</i>	Memberi tanda akhir pada <i>listbox</i> , status pada <i>status bar</i> terganti menjadi <i>done</i> dan <i>ribbon page</i> tempo dan <i>generate enable</i>	Tanda akhir pada <i>listbox</i> ditambahkan, status pada <i>status bar</i> terganti menjadi <i>done</i> dan <i>ribbon page</i> tempo dan <i>generate enable</i>	OK

Tabel 4.3 Tabel pengujian tombol-tombol dan *form* not angka4. Pengujian *form* notasi angka *generate*

No	Aksi	Hasil yang diharapkan	Output	Ket
1	Form show	Button <i>generate</i> not angka <i>enable</i>	Button <i>generate</i> not angka <i>enable</i>	OK
2	Form close	Button <i>generate</i> not angka <i>disable</i>	Button <i>generate</i> not angka <i>disable</i>	OK

Tabel 4.4 Tabel pengujian *form* notasi angka *generate*

5. Pengujian *form* notasi balok

No	Aksi	Hasil yang diharapkan	<i>Output</i>	Ket
1	Form show	Button <i>generate</i> not balok <i>enable</i>	Button <i>generate</i> not balok <i>enable</i>	OK
2	Form close	Button <i>generate</i> not balok <i>disnable</i>	Button <i>generate</i> not balok <i>disnable</i>	OK

Tabel 4.5 Tabel pengujian *form* notasi balok

BAB 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Setelah melalui seluruh tahap dan proses dalam pembuatan perangkat lunak konversi notasi angka ke dalam notasi balok ini, maka dapat ditarik beberapa kesimpulan sebagai berikut, yaitu :

1. Perancangan perangkat lunak konversi notasi angka ke notasi balok menggunakan *tools Visual Studio 2012* menggunakan komponen-komponen dari *Devexpress* sebagai komponen untuk membangun *User Interface*. Partitur notasi angka maupun notasi balok dibuat berdasarkan data-data dari *text file* yang dibuat oleh pengguna maupun pengambilan data yang sudah tersimpan sebelumnya. Hasil keluaran dari perangkat lunak adalah *file gambar berextensi jpeg*.
2. Memainkan nada-nada untuk analisa dilakukan dengan cara memanggil *system sound* pada *windows* dengan beberapa parameter yang dibutuhkan, yaitu frekuensi dan durasi yang disesuaikan dengan notasi notasi yang telah dibuat.

5.2 Saran

Saran yang dapat berikan untuk pengembangan perangkat lunak tugas akhir ini agar lebih baik adalah sebagai berikut :

1. Menambahkan tanda-tanda notasi seperti *tie, pugar, mol, crescendo, decresendo, chord* dan yang lain nya agar dapat memperluas pembuatan partitur.
2. Menambahkan jumlah birama, tempo dan tada kunci sesuai kebutuhan pengguna.
3. Menambahkan fungsi *leggato* dalam memainkan nada-nada yang telah dibuat oleh pengguna.

DAFTAR PUSTAKA

- Banoe, Pono. 2003. *Kamus Musik*. Yogyakarta: Kanisius.
- Binanto, Iwan. 2005. *Konsep Dasar Program*. Jakarta: PT. Elex Media Komputindo.
- Kadir, Abdul. 2009. *Dasar Perancangan dan Implementasi Database Relasional*, Edisi I. Yogyakarta : Andi.
- Kurniadi, A. 2005. *Pemrograman Microsoft Visual Basic 6*. Jakarta: Elex Media Komputindo.
- Madcoms. 2008. *Microsoft Visual Basic 6.0*. Yogyakarta : C.V Andi
- Mulyanto, Eko S. 2007. *Panduan Dasar Bermain Jazz*. Jakarta: Kawan Pustaka.
- Nickol, Peter. 2005. *Panduan Praktis Membaca Notasi Musik*. Jakarta: PT. Gramedia Pustaka Utama.
- Pandia, H. 2006. *Pemrograman dengan Visual Basic*. Jakarta: Penerbit Erlangga
- Presman, Roger S. 2010. *Software Engineering A Practitioner's Approach Seventh Edition*. New York: McGraw-Hill Companies.inc
- RZ, Aklat. 2014. *Reading Music Notation*
- Simarmata, Janner. 2010. *Rekayasa Perangkat Lunak*. Yogyakarta: Andi.
- Sugiyono. 2005. *Pemrograman Terstruktur*. Kuningan: Panju Gumilang Press.
- Surmani, Andrew dan Karen Farnum Surmani, Morton Manus. 2004. *Alfred's Essentials of Music Theory*. Alfred Music Publishing.
- <https://msdn.microsoft.com/en-us/library>, diakses tanggal 7 Januari 2015
- <http://visualbasic.freetutes.com>, diakses tanggal 11 Desember 2014
- <https://www.devexpress.com/Support/Documentation>, diakses tanggal 8 Januari 2015
- <http://www.hit.ac.il/staff/leonidM/information-systems/ch63.html>, diakses tanggal 10 November 2014
- <http://www.musicarrangers.com/star-theory>, diakses tanggal 2 November 2014
- <http://www.musictheory.net>, diakses tanggal 2 November 2014
- <http://www.viola.com>, diakses tanggal 2 November 2014

LAMPIRAN LISTING PROGRAM


```

Imports System.ComponentModel
Imports System.Text
Imports DevExpress.XtraTabbedMdi
Imports DevExpress.XtraTab
Imports DevExpress.LookAndFeel
Imports System.IO
Imports System.Drawing.Graphics

Partial Public Class FrmMain
    Inherits DevExpress.XtraBars.Ribbon.RibbonForm

    Dim x As Integer = 20
    Dim y As Integer = 10

    Dim txtTitle As New System.Drawing.Font("calibri", 4, FontStyle.Bold)
    Dim txtSubTitle As New System.Drawing.Font("calibri", 3,
FontStyle.Bold)
    Dim txtNot As New System.Drawing.Font("calibri", 2, FontStyle.Regular)
    Dim txtDot As New System.Drawing.Font("calibri", 2, FontStyle.Bold)

    Shared Sub New()
        UserLookAndFeel.Default.SetSkinStyle("Office 2013")
        DevExpress.UserSkins.BonusSkins.Register()
        DevExpress.Skins.SkinManager.EnableFormSkins()
    End Sub

    Public Sub New()
        InitializeComponent()
    End Sub

    Private mdiManager As XtraTabbedMdiManager = New XtraTabbedMdiManager()

    Private Sub initialize()
        txtTitle.EditValue = ""
        txtAuthor.EditValue = ""
        txtBar.EditValue = repBar.Items(0)
        txtScale.EditValue = repScale.Items(0)
        cmbTempo.EditValue = repTempo.Items(0)
    End Sub

    Public Sub initstatusbar()
        bartxtAuthor.Caption = "Author"
        bartxtBar.Caption = "Bar"
        bartxtScale.Caption = "Scale"
        bartxtTitle.Caption = "Title"
        bartxtStatus.Caption = "New Partitur"
    End Sub

    Public Sub cekstatus()
        If bartxtStatus.Caption = "Done" Then
            rpgGenerate.Enabled = True
            rpgSound.Enabled = True
        End If
    End Sub

    Public Sub mdiparentlocation()
        frmFile.MdiParent = Me
        frmNotAngka.MdiParent = Me
        frmNotBalok.MdiParent = Me
        frmNotAngkaGenerate.MdiParent = Me
    End Sub
End Class

```

```

    frmFile.Text = "File Partitur"
    frmNotAngka.Text = "Notasi Angka"
    frmNotBalok.Text = "Notasi Balok"
    frmNotAngkaGenerate.Text = "Notasi Angka"
End Sub

Private Sub FrmMain_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    mdiManager.AppearancePage.HeaderActive.Font = New
System.Drawing.Font("Calibri", 11, System.Drawing.FontStyle.Bold)
    mdiManager.MdiParent = Me
    RadialMenu.Manager = BarManager
    rpgNew.Enabled = False
    rpgSound.Enabled = False
    rpgGenerate.Enabled = False
    btnExportNotAngka.Enabled = False
    btnExportNotBalok.Enabled = False
    initialize()
    initstatusbar()
    mdiparentlocation()
End Sub

Private Sub btnNew_ItemClick(sender As Object, e As
DevExpress.XtraBars.ItemClickEventArgs) Handles btnNew.ItemClick
    rpgNew.Enabled = True
    rpgFile.Enabled = False
End Sub

Private Sub btnCreate_ItemClick(sender As Object, e As
DevExpress.XtraBars.ItemClickEventArgs) Handles brnCreate.ItemClick
    frmFile.lsbFile.Items.Clear()

    If txtTitle.EditValue.ToString <> "" And
txtAuthor.EditValue.ToString <> "" Then
        Try
            frmFile.lsbFile.Items.Add(txtTitle.EditValue.ToString + ":"
+ txtAuthor.EditValue.ToString + ":" + txtBar.EditValue.ToString _
+ ":" +
txtScale.EditValue.ToString + ":")

            bartxtTitle.Caption = "Title : " +
txtTitle.EditValue.ToString
            bartxtAuthor.Caption = "Author : " +
txtAuthor.EditValue.ToString
            bartxtBar.Caption = txtBar.EditValue.ToString
            bartxtScale.Caption = txtScale.EditValue.ToString

            Dim bmp As Image = My.Resources.LayerNot
            Using g As Graphics = Graphics.FromImage(bmp)
                g.DrawString(txtTitle.EditValue.ToString, TextTitle,
Brushes.Black, 10, 10)
                g.DrawString(txtAuthor.EditValue.ToString,
TextSubTitle, Brushes.Black, 10, 25)
                g.DrawString(txtBar.EditValue.ToString, TextSubTitle,
Brushes.Black, 10, 35)
                g.DrawString(txtScale.EditValue.ToString, TextSubTitle,
Brushes.Black, 10, 45)
            End Using

```

```

        frmNotAngka.picNotAngka.Image = bmp
        mdiparentlocation()
        frmFile.Show()
        frmNotAngka.Refresh()
        frmNotAngka.Show()
        frmNotAngkaGenerate.Close()
        frmNotBalok.Close()
        initialize()
        rpgNew.Enabled = False
        rpgFile.Enabled = True
        bartxtStatus.Caption = "Edit"

    Catch ex As Exception
        MessageBox.Show(ex.Message)
    End Try
Else
    MessageBox.Show("Data is empty.", "Warning")
End If

End Sub

Private Sub btnOpen_ItemClick(sender As Object, e As
DevExpress.XtraBars.ItemClickEventArgs) Handles btnOpen.ItemClick
    Dim myStream As Stream = Nothing
    Dim ofd As New OpenFileDialog()

    ofd.Title = "Open Partitur"
    ofd.InitialDirectory = "D:\\"
    ofd.Filter = "txt files (*.txt)|*.txt"
    ofd.FilterIndex = 0
    ofd.RestoreDirectory = True

    If ofd.ShowDialog() = System.Windows.Forms.DialogResult.OK Then
        Try
            myStream = ofd.OpenFile()
            If (myStream IsNot Nothing) Then
                frmFile.lsbFile.Items.Clear()

                frmFile.lsbFile.Items.AddRange(IO.File.ReadAllLines(ofd.FileName))

                Dim a As Integer = frmFile.lsbFile.Items.Count - 1
                If frmFile.lsbFile.Items.Item(a).ToString = "ENDPART:"
Then
                    frmFile.Show()
                    bartxtStatus.Caption = "Done"
                    cekstatus()
                Else
                    MessageBox.Show("Incorrect File")
                    frmFile.lsbFile.Items.Clear()
                    frmFile.Close()
                End If
            End If
        Catch ex As Exception
            MessageBox.Show("Cannot read file from disk. Original
error: " & ex.Message)
        Finally
            If (myStream IsNot Nothing) Then
                myStream.Close()
            End If
        End Try
    End If
End Sub

```

```

        mdiparentlocation()

        frmNotAngkaGenerate.Close()
        frmNotBalok.Close()
        frmNotAngka.Close()
    End Try
End If
End Sub

Private Sub btnSave_ItemClick(sender As Object, e As
DevExpress.XtraBars.ItemClickEventArgs) Handles btnSave.ItemClick
    Dim sfd As New SaveFileDialog()

    sfd.Title = "Save Partitur"
    sfd.InitialDirectory = "D:\\"
    sfd.Filter = "txt files (*.txt)|*.txt"
    sfd.FilterIndex = 0
    sfd.RestoreDirectory = True

    If frmFile.lsbFile.Items.Count <> 0 Then
        Try
            If sfd.ShowDialog() = DialogResult.OK Then
                Using writer = New StreamWriter(sfd.FileName)
                    For Each o As Object In frmFile.lsbFile.Items
                        writer.WriteLine(o)
                    Next
                End Using
            End If
        Catch ex As Exception
            MessageBox.Show(ex.Message)
        End Try
    End If
End Sub

Private Sub btnGenerateAngka_ItemClick(sender As Object, e As
DevExpress.XtraBars.ItemClickEventArgs) Handles btnGenerateAngka.ItemClick
    mdiparentlocation()
    frmNotAngkaGenerate.Show()
    frmNotAngka.Close()
    frmNotAngkaGenerate.picAngkaGenerate.Image = Nothing
    frmFile.drawnotangka()
End Sub

Private Sub btnGenerateBalok_ItemClick(sender As Object, e As
DevExpress.XtraBars.ItemClickEventArgs) Handles btnGenerateBalok.ItemClick
    mdiparentlocation()
    frmNotBalok.Show()
    frmNotAngka.Close()
    frmNotBalok.picNotBalok.Image = Nothing
    frmFile.drawnotbalok()
End Sub

Private Sub btnGenerate_ItemClick(sender As Object, e As
DevExpress.XtraBars.ItemClickEventArgs) Handles btnGenerate.ItemClick
    Me.Cursor = New Cursor(Cursor.Current.Handle)
    Dim pt As Point = New Point(Cursor.Position.X - 50,
Cursor.Position.Y - 50)
    RadialMenu.ShowPopup(pt)
End Sub

```

```

Private Sub btnPlay_ItemClick(sender As Object, e As
DevExpress.XtraBars.ItemClickEventArgs) Handles btnPlay.ItemClick
    frmFile.playmusic(cmbTempo.EditValue.ToString)
End Sub

Private Sub btnExportNotAngka_ItemClick(sender As Object, e As
DevExpress.XtraBars.ItemClickEventArgs) Handles btnExportNotAngka.ItemClick
    Dim sfd As New SaveFileDialog()

    sfd.Title = "Export partitur angka"
    sfd.InitialDirectory = "D:\\"
    sfd.Filter = "jpeg (*.jpeg)|*.jpeg"
    sfd.FilterIndex = 0
    sfd.RestoreDirectory = True

    Try
        If sfd.ShowDialog() = DialogResult.OK Then

frmNotAngkaGenerate.picAngkaGenerate.Image.Save(sfd.FileName,
Imaging.ImageFormat.Jpeg)
            End If
        Catch ex As Exception
            MessageBox.Show(ex.Message)
        End Try

        sfd.Dispose()

    End Sub

Private Sub btnExportNotBalok_ItemClick(sender As Object, e As
DevExpress.XtraBars.ItemClickEventArgs) Handles btnExportNotBalok.ItemClick
    Dim sfd As New SaveFileDialog()

    sfd.Title = "Export partitur balok"
    sfd.InitialDirectory = "D:\\"
    sfd.Filter = "jpeg (*.jpeg)|*.jpeg"
    sfd.FilterIndex = 0
    sfd.RestoreDirectory = True

    Try
        If sfd.ShowDialog() = DialogResult.OK Then
            frmNotBalok.picNotBalok.Image.Save(sfd.FileName,
Imaging.ImageFormat.Jpeg)
            End If
        Catch ex As Exception
            MessageBox.Show(ex.Message)
        End Try

        sfd.Dispose()
    End Sub
End Class

```

```

Imports System.Drawing.Graphics
Imports DevExpress.XtraBars
Imports DevExpress.XtraBars.Ribbon
Imports System.Threading

Public Class frmFile
    Dim kata, nadadasar As String
    Dim partarray() As String
    Dim x, y, xstart, ystart, jumlahbaris As Integer
    Dim TextTitle As New System.Drawing.Font("calibri", 4, FontStyle.Bold)
    Dim TextSubTitle As New System.Drawing.Font("calibri", 3,
FontStyle.Bold)
    Dim TextNot As New System.Drawing.Font("calibri", 3, FontStyle.Regular)
    Dim TextDot As New System.Drawing.Font("calibri", 3, FontStyle.Bold)
    Dim bmp As Image = My.Resources.LayerNot

    'set item file from array
    Public Sub setarray()
        Erase partarray
        kata = ""

        For i As Integer = 0 To lsbFile.Items.Count - 1
            kata += lsbFile.Items.Item(i).ToString
        Next
        partarray = kata.Split(":")
    End Sub

    Public Sub getitemfile()
        setarray()
        jumlahbaris = 0
        Dim arrayline = UBound(partarray)

        For i As Integer = 1 To arrayline
            If partarray(i) = "END" Then
                jumlahbaris += 1
            End If
        Next
    End Sub

    'draw not angka
    Public Sub cekrange(tmprange As String, tmpnot As String, tmpsgn As
String)
        If tmprange = "T" Then
            If tmpsgn = "N" Then
                Using g As Graphics = Graphics.FromImage(bmp)
                    g.DrawString(tmpnot, TextNot, Brushes.Black, x, y)
                    g.DrawString(".", TextDot, Brushes.Black, x + 1, y - 9)
                End Using
            Else
                Using g As Graphics = Graphics.FromImage(bmp)
                    g.DrawString(tmpnot, TextNot, Brushes.Black, x, y)
                    g.DrawString(".", TextDot, Brushes.Black, x + 1, y - 9)
                    g.DrawLine(Pens.Black, x, y + 10, x + 7, y + 3)
                End Using
            End If
        ElseIf tmprange = "R" Then
            If tmpsgn = "N" Then
                Using g As Graphics = Graphics.FromImage(bmp)
                    g.DrawString(tmpnot, TextNot, Brushes.Black, x, y)

```

```

        g.DrawString(".", TextDot, Brushes.Black, +1, y + 9)
    End Using
Else
    Using g As Graphics = Graphics.FromImage(bmp)
        g.DrawString(tmpnot, TextNot, Brushes.Black, x, y)
        g.DrawString(".", TextDot, Brushes.Black, +1, y + 9)
        g.DrawLine(Pens.Black, x, y + 10, x + 7, y + 3)
    End Using
End If

ElseIf tmprange = "N" Then
    If tmpsgn = "N" Then
        Using g As Graphics = Graphics.FromImage(bmp)
            g.DrawString(tmpnot, TextNot, Brushes.Black, x, y)
        End Using
    Else
        Using g As Graphics = Graphics.FromImage(bmp)
            g.DrawString(tmpnot, TextNot, Brushes.Black, x, y)
            g.DrawLine(Pens.Black, x, y + 10, x + 7, y + 3)
        End Using
    End If
End If
End Sub

Public Sub drawnotangka()
    setarray()
    bmp = My.Resources.LayerNot
    x = 10
    y = 95

    Using g As Graphics = Graphics.FromImage(bmp)
        g.DrawString(partarray(0).ToString, TextTitle, Brushes.Black,
10, 10)
        g.DrawString(partarray(1).ToString, TextSubTitle,
Brushes.Black, 10, 30)
        g.DrawString(partarray(2).ToString, TextSubTitle,
Brushes.Black, 10, 45)
        g.DrawString(partarray(3).ToString, TextSubTitle,
Brushes.Black, 10, 60)
    End Using

    Dim arraycount = UBound(partarray)

    For i As Integer = 4 To arraycount - 1 Step 1

        If partarray(i) <> "ENDPART" Then
            If partarray(i) <> "END" Then

                If partarray(i) = "B" Then
                    Using g As Graphics = Graphics.FromImage(bmp)
                        g.DrawLine(Pens.Black, x, y + 10, x, y - 10)
                    End Using
                    x += 20

                ElseIf partarray(i) <> "B" Then
                    Dim temp As String = partarray(i).Substring(0, 1)

                    If temp = "W" Then
                        Dim t As String = partarray(i).Substring(1, 1)
                        Dim tn As String = partarray(i).Substring(2, 1)

```

```

Dim ts As String = partarray(i).Substring(3, 1)
cekrange(t, tn, ts)
Using g As Graphics = Graphics.FromImage bmp)
    g.DrawString(".", TextDot, Brushes.Black, x
+ 20, y)
    g.DrawString(".", TextDot, Brushes.Black, x
+ 40, y)
    g.DrawString(".", TextDot, Brushes.Black, x
+ 60, y)
End Using
x += 80
ElseIf temp = "H" Then
Dim t As String = partarray(i).Substring(1, 1)
Dim tn As String = partarray(i).Substring(2, 1)
Dim ts As String = partarray(i).Substring(3, 1)
cekrange(t, tn, ts)
Using g As Graphics = Graphics.FromImage bmp)
    g.DrawString(".", TextDot, Brushes.Black, x
+ 20, y)
End Using
x += 40
ElseIf temp = "Q" Then
Dim t As String = partarray(i).Substring(1, 1)
Dim tn As String = partarray(i).Substring(2, 1)
Dim ts As String = partarray(i).Substring(3, 1)
cekrange(t, tn, ts)
x += 20
ElseIf temp = "E" Then
Dim t As String = partarray(i).Substring(1, 1)
Dim tn As String = partarray(i).Substring(2, 1)
Dim ts As String = partarray(i).Substring(3, 1)
Dim t1 As String = partarray(i).Substring(4, 1)
Dim tn1 As String = partarray(i).Substring(5,
1)
Dim ts1 As String = partarray(i).Substring(6,
1)
cekrange(t, tn, ts)
x += 20
cekrange(t1, tn1, ts1)
Using g As Graphics = Graphics.FromImage bmp)
    g.DrawLine(Pens.Black, x - 20, y - 10, x +
5, y - 10)
End Using
x += 20
ElseIf temp = "S" Then
Dim t As String = partarray(i).Substring(1, 1)
Dim tn As String = partarray(i).Substring(2, 1)
Dim ts As String = partarray(i).Substring(3, 1)
Dim t1 As String = partarray(i).Substring(4, 1)
Dim tn1 As String = partarray(i).Substring(5,
1)
Dim ts1 As String = partarray(i).Substring(6,
1)
Dim t2 As String = partarray(i).Substring(7, 1)
Dim tn2 As String = partarray(i).Substring(8,
1)
Dim ts2 As String = partarray(i).Substring(9,
1)
Dim t3 As String = partarray(i).Substring(10,
1)

```



```

1) Dim tn3 As String = partarray(i).Substring(11,
1) Dim ts3 As String = partarray(i).Substring(12,

    cekrange(t, tn, ts)
    x += 20
    cekrange(t1, tn1, ts1)
    x += 20
    cekrange(t2, tn2, ts2)
    x += 20
    cekrange(t3, tn3, ts3)
    Using g As Graphics = Graphics.FromImage bmp)
        g.DrawLine(Pens.Black, x - 60, y - 10, x +
5, y - 10)

        End Using
        x += 20
    End If
    End If
    ElseIf partarray(i) = "END" Then
        x = 10
        y += 35
    End If
Else
    MessageBox.Show("Generate Done", "Sukses!")
End If
Next

    frmNotAngkaGenerate.picAngkaGenerate.Image = bmp
End Sub

'draw not balok
Public Sub cekxstart()
    nadadasar = FrmMain.bartxtScale.Caption

    If nadadasar = "C" Then
        xstart = 60
    ElseIf nadadasar = "G" Then
        xstart = 70
    ElseIf nadadasar = "D" Then
        xstart = 80
    ElseIf nadadasar = "A" Then
        xstart = 90
    ElseIf nadadasar = "E" Then
        xstart = 100
    ElseIf nadadasar = "B" Then
        xstart = 110
    ElseIf nadadasar = "F" Then
        xstart = 120
    End If
End Sub

Public Sub cekystart(temp As Integer)
    nadadasar = FrmMain.bartxtScale.Caption

    If nadadasar = "C" Then
        ystart = 125 + temp
    ElseIf nadadasar = "G" Then
        ystart = 105 + temp
    ElseIf nadadasar = "D" Then

```

```

        ystart = 120 + temp
    ElseIf nadadasar = "A" Then
        ystart = 100 + temp
    ElseIf nadadasar = "E" Then
        ystart = 115 + temp
    ElseIf nadadasar = "B" Then
        ystart = 95 + temp
    ElseIf nadadasar = "F" Then
        ystart = 110 + temp
    End If
End Sub

Function getyposition(tmprange As String, tmpnot As String) As Integer
    If tmprange = "T" Then
        Select Case tmpnot
            Case "1"
                Return ystart - 35
            Case "2"
                Return ystart - 40
            Case "3"
                Return ystart - 45
            Case "4"
                Return ystart - 50
            Case "5"
                Return ystart - 55
            Case "6"
                Return ystart - 60
            Case "7"
                Return ystart - 65
        End Select
    ElseIf tmprange = "N" Then
        Select Case tmpnot
            Case "1"
                Return ystart
            Case "2"
                Return ystart - 5
            Case "3"
                Return ystart - 10
            Case "4"
                Return ystart - 15
            Case "5"
                Return ystart - 20
            Case "6"
                Return ystart - 25
            Case "7"
                Return ystart - 30
        End Select
    ElseIf tmprange = "R" Then
        Select Case tmpnot
            Case "1"
                Return ystart + 35
            Case "2"
                Return ystart + 30
            Case "3"
                Return ystart + 25
            Case "4"
                Return ystart + 20
            Case "5"
                Return ystart + 15
            Case "6"

```

```

        Return ystart + 10
    Case "7"
        Return ystart + 5
    End Select
End If

End Function

Public Sub cekrangebalok(tmprange As String, tmpnot As String, tmpimg
As Image)
    Dim tmp As Integer = 0

    If tmprange = "T" Then
        Select Case tmpnot
            Case "1"
                Using g As Graphics = Graphics.FromImage bmp
                    g.DrawImage(tmpimg, xstart, ystart - 35)
                    tmp = ystart - 35
                End Using
            Case "2"
                Using g As Graphics = Graphics.FromImage bmp
                    g.DrawImage(tmpimg, xstart, ystart - 40)
                    tmp = ystart - 40
                End Using
            Case "3"
                Using g As Graphics = Graphics.FromImage bmp
                    g.DrawImage(tmpimg, xstart, ystart - 45)
                    tmp = ystart - 45
                End Using
            Case "4"
                Using g As Graphics = Graphics.FromImage bmp
                    g.DrawImage(tmpimg, xstart, ystart - 50)
                    tmp = ystart - 50
                End Using
            Case "5"
                Using g As Graphics = Graphics.FromImage bmp
                    g.DrawImage(tmpimg, xstart, ystart - 55)
                    tmp = ystart - 55
                End Using
            Case "6"
                Using g As Graphics = Graphics.FromImage bmp
                    g.DrawImage(tmpimg, xstart, ystart - 60)
                    tmp = ystart - 60
                End Using
            Case "7"
                Using g As Graphics = Graphics.FromImage bmp
                    g.DrawImage(tmpimg, xstart, ystart - 65)
                    tmp = ystart - 65
                End Using
        End Select
    ElseIf tmprange = "N" Then
        Select Case tmpnot
            Case "1"
                Using g As Graphics = Graphics.FromImage bmp
                    g.DrawImage(tmpimg, xstart, ystart)
                    tmp = ystart
                End Using
            Case "2"
                Using g As Graphics = Graphics.FromImage bmp
                    g.DrawImage(tmpimg, xstart, ystart - 5)

```

```

        tmp = ystart - 5
    End Using
Case "3"
    Using g As Graphics = Graphics.FromImage bmp
        g.DrawImage(tmpimg, xstart, ystart - 10)
        tmp = ystart - 10
    End Using
Case "4"
    Using g As Graphics = Graphics.FromImage bmp
        g.DrawImage(tmpimg, xstart, ystart - 15)
        tmp = ystart - 15
    End Using
Case "5"
    Using g As Graphics = Graphics.FromImage bmp
        g.DrawImage(tmpimg, xstart, ystart - 20)
        tmp = ystart - 20
    End Using
Case "6"
    Using g As Graphics = Graphics.FromImage bmp
        g.DrawImage(tmpimg, xstart, ystart - 25)
        tmp = ystart - 25
    End Using
Case "7"
    Using g As Graphics = Graphics.FromImage bmp
        g.DrawImage(tmpimg, xstart, ystart - 30)
        tmp = ystart - 30
    End Using
End Select
ElseIf tmprange = "R" Then
    Select Case tmpnot
    Case "1"
        Using g As Graphics = Graphics.FromImage bmp
            g.DrawImage(tmpimg, xstart, ystart + 35)
            tmp = ystart + 35
        End Using
    Case "2"
        Using g As Graphics = Graphics.FromImage bmp
            g.DrawImage(tmpimg, xstart, ystart + 30)
            tmp = ystart + 30
        End Using
    Case "3"
        Using g As Graphics = Graphics.FromImage bmp
            g.DrawImage(tmpimg, xstart, ystart + 25)
            tmp = ystart + 25
        End Using
    Case "4"
        Using g As Graphics = Graphics.FromImage bmp
            g.DrawImage(tmpimg, xstart, ystart + 20)
            tmp = ystart + 20
        End Using
    Case "5"
        Using g As Graphics = Graphics.FromImage bmp
            g.DrawImage(tmpimg, xstart, ystart + 15)
            tmp = ystart + 15
        End Using
    Case "6"
        Using g As Graphics = Graphics.FromImage bmp
            g.DrawImage(tmpimg, xstart, ystart + 10)
            tmp = ystart + 10
        End Using
    
```

```

        Case "7"
            Using g As Graphics = Graphics.FromImage(bmp)
                g.DrawImage(tmpimg, xstart, ystart + 5)
                tmp = ystart + 5
            End Using
        End Select
    End If

    If tmp <= y - 35 Then
        Dim i As Double = 0
        i = (tmp - (y - 35)) / 5

        If i = 0 Or i = -1 Then
            Using g As Graphics = Graphics.FromImage(bmp)
                g.DrawLine(Pens.Black, xstart - 2, ystart - 25, xstart
+ 17, ystart - 25)
            End Using
        ElseIf i = -2 Or i = -3 Then
            Using g As Graphics = Graphics.FromImage(bmp)
                g.DrawLine(Pens.Black, xstart - 2, ystart - 25, xstart
+ 17, ystart - 25)
                g.DrawLine(Pens.Black, xstart - 2, ystart - 35, xstart
+ 17, ystart - 35)
            End Using
        ElseIf i = -4 Or i = -5 Then
            Using g As Graphics = Graphics.FromImage(bmp)
                g.DrawLine(Pens.Black, xstart - 2, ystart - 25, xstart
+ 17, ystart - 25)
                g.DrawLine(Pens.Black, xstart - 2, ystart - 35, xstart
+ 17, ystart - 35)
                g.DrawLine(Pens.Black, xstart - 2, ystart - 45, xstart
+ 17, ystart - 45)
            End Using
        ElseIf i = -6 Or i = -7 Then
            Using g As Graphics = Graphics.FromImage(bmp)
                g.DrawLine(Pens.Black, xstart - 2, ystart - 25, xstart
+ 17, ystart - 25)
                g.DrawLine(Pens.Black, xstart - 2, ystart - 35, xstart
+ 17, ystart - 35)
                g.DrawLine(Pens.Black, xstart - 2, ystart - 45, xstart
+ 17, ystart - 45)
                g.DrawLine(Pens.Black, xstart - 2, ystart - 55, xstart
+ 17, ystart - 55)
            End Using
        End If

        ElseIf tmp >= y + 25 Then
            Dim i As Double = 0
            i = (tmp - (y + 25)) / 5

            If i = 0 Or i = 1 Then
                Using g As Graphics = Graphics.FromImage(bmp)
                    g.DrawLine(Pens.Black, xstart - 2, ystart + 25, xstart
+ 17, ystart + 25)
                End Using
            ElseIf i = 2 Or i = 3 Then
                Using g As Graphics = Graphics.FromImage(bmp)
                    g.DrawLine(Pens.Black, xstart - 2, ystart + 25, xstart
+ 17, ystart + 25)
                End Using
            End If
        End If
    End If

```

```

        g.DrawLine(Pens.Black, xstart - 2, ystart + 35, xstart
+ 17, ystart + 35)
    End Using
    ElseIf i = 4 Or i = 5 Then
        Using g As Graphics = Graphics.FromImage bmp)
            g.DrawLine(Pens.Black, xstart - 2, ystart + 25, xstart
+ 17, ystart + 25)
            g.DrawLine(Pens.Black, xstart - 2, ystart + 35, xstart
+ 17, ystart + 35)
            g.DrawLine(Pens.Black, xstart - 2, ystart + 45, xstart
+ 17, ystart + 45)
        End Using
    ElseIf i = 6 Or i = 7 Then
        Using g As Graphics = Graphics.FromImage bmp)
            g.DrawLine(Pens.Black, xstart - 2, ystart + 25, xstart
+ 17, ystart + 25)
            g.DrawLine(Pens.Black, xstart - 2, ystart + 35, xstart
+ 17, ystart + 35)
            g.DrawLine(Pens.Black, xstart - 2, ystart + 45, xstart
+ 17, ystart + 45)
            g.DrawLine(Pens.Black, xstart - 2, ystart + 55, xstart
+ 17, ystart + 55)
        End Using
    End If

End If

End Sub

Public Sub drawline()
    Dim imgClef As Image = My.Resources.Clef
    Dim imgcress As Image = My.Resources.Cress
    nadadasar = FrmMain.bartxtScale.Caption

    Select Case nadadasar
        Case "C"
            Using g As Graphics = Graphics.FromImage bmp)
                g.DrawImage(imgClef, x, y - 13)
                g.DrawLine(Pens.Black, x, y, x + 1250, y)
                g.DrawLine(Pens.Black, x, y + 10, x + 1250, y + 10)
                g.DrawLine(Pens.Black, x, y + 20, x + 1250, y + 20)
                g.DrawLine(Pens.Black, x, y + 30, x + 1250, y + 30)
                g.DrawLine(Pens.Black, x, y + 40, x + 1250, y + 40)
                'xstart = 60
            End Using
        Case "G"
            Using g As Graphics = Graphics.FromImage bmp)
                g.DrawImage(imgClef, x, y - 13)
                g.DrawLine(Pens.Black, x, y, x + 1250, y)
                g.DrawLine(Pens.Black, x, y + 10, x + 1250, y + 10)
                g.DrawLine(Pens.Black, x, y + 20, x + 1250, y + 20)
                g.DrawLine(Pens.Black, x, y + 30, x + 1250, y + 30)
                g.DrawLine(Pens.Black, x, y + 40, x + 1250, y + 40)
                g.DrawImage(imgcress, x + 40, y - 7)
                'xstart = 70
            End Using
        Case "D"
            Using g As Graphics = Graphics.FromImage bmp)
                g.DrawImage(imgClef, x, y - 13)
                g.DrawLine(Pens.Black, x, y, x + 1250, y)
            End Using
    End Select
End Sub

```

```

        g.DrawLine(Pens.Black, x, y + 10, x + 1250, y + 10)
        g.DrawLine(Pens.Black, x, y + 20, x + 1250, y + 20)
        g.DrawLine(Pens.Black, x, y + 30, x + 1250, y + 30)
        g.DrawLine(Pens.Black, x, y + 40, x + 1250, y + 40)
        g.DrawImage(imgcress, x + 40, y - 7)
        g.DrawImage(imgcress, x + 50, y + 10)
        'xstart = 80
    End Using
Case "A"
    Using g As Graphics = Graphics.FromImage(bmp)
        g.DrawImage(imgClef, x, y - 13)
        g.DrawLine(Pens.Black, x, y, x + 1250, y)
        g.DrawLine(Pens.Black, x, y + 10, x + 1250, y + 10)
        g.DrawLine(Pens.Black, x, y + 20, x + 1250, y + 20)
        g.DrawLine(Pens.Black, x, y + 30, x + 1250, y + 30)
        g.DrawLine(Pens.Black, x, y + 40, x + 1250, y + 40)
        g.DrawImage(imgcress, x + 40, y - 7)
        g.DrawImage(imgcress, x + 50, y + 10)
        g.DrawImage(imgcress, x + 60, y - 10)
        'xstart = 90
    End Using
Case "E"
    Using g As Graphics = Graphics.FromImage(bmp)
        g.DrawImage(imgClef, x, y - 13)
        g.DrawLine(Pens.Black, x, y, x + 1250, y)
        g.DrawLine(Pens.Black, x, y + 10, x + 1250, y + 10)
        g.DrawLine(Pens.Black, x, y + 20, x + 1250, y + 20)
        g.DrawLine(Pens.Black, x, y + 30, x + 1250, y + 30)
        g.DrawLine(Pens.Black, x, y + 40, x + 1250, y + 40)
        g.DrawImage(imgcress, x + 40, y - 7)
        g.DrawImage(imgcress, x + 50, y + 10)
        g.DrawImage(imgcress, x + 60, y - 10)
        g.DrawImage(imgcress, x + 70, y + 2)
        'xstart = 100
    End Using
Case "B"
    Using g As Graphics = Graphics.FromImage(bmp)
        g.DrawImage(imgClef, x, y - 13)
        g.DrawLine(Pens.Black, x, y, x + 1250, y)
        g.DrawLine(Pens.Black, x, y + 10, x + 1250, y + 10)
        g.DrawLine(Pens.Black, x, y + 20, x + 1250, y + 20)
        g.DrawLine(Pens.Black, x, y + 30, x + 1250, y + 30)
        g.DrawLine(Pens.Black, x, y + 40, x + 1250, y + 40)
        g.DrawImage(imgcress, x + 40, y - 7)
        g.DrawImage(imgcress, x + 50, y + 10)
        g.DrawImage(imgcress, x + 60, y - 10)
        g.DrawImage(imgcress, x + 70, y + 2)
        g.DrawImage(imgcress, x + 80, y + 20)
        'xstart = 110
    End Using
Case "F"
    Using g As Graphics = Graphics.FromImage(bmp)
        g.DrawImage(imgClef, x, y - 13)
        g.DrawLine(Pens.Black, x, y, x + 1250, y)
        g.DrawLine(Pens.Black, x, y + 10, x + 1250, y + 10)
        g.DrawLine(Pens.Black, x, y + 20, x + 1250, y + 20)
        g.DrawLine(Pens.Black, x, y + 30, x + 1250, y + 30)
        g.DrawLine(Pens.Black, x, y + 40, x + 1250, y + 40)
        g.DrawImage(imgcress, x + 40, y - 7)
        g.DrawImage(imgcress, x + 50, y + 10)
    
```

```

        g.DrawImage(imgcress, x + 60, y - 10)
        g.DrawImage(imgcress, x + 70, y + 2)
        g.DrawImage(imgcress, x + 80, y + 20)
        g.DrawImage(imgcress, x + 90, y)
        'xstart = 120
    End Using
End Select

End Sub

Public Sub drawnotbalok()
    Dim tmpy As Integer = 0
    Dim imgwholenote As Image = My.Resources.Whole_Not
    Dim imghalfnote As Image = My.Resources.Half_not
    Dim imgquaternote As Image = My.Resources.Quarter_Not
    Dim imgeightnote As Image = My.Resources.eighth_not
    Dim imgsixteenthnote As Image = My.Resources.sixteenth_note
    Dim imgwholerest As Image = My.Resources.restwhole
    Dim imghalfrest As Image = My.Resources.resthalf
    Dim imgquaterrest As Image = My.Resources.restquarter
    Dim imgeightrest As Image = My.Resources.resteighth
    Dim imgsixteentrest As Image = My.Resources.restsixteenth

    getitemfile()
    bmp = My.Resources.LayerNot
    y = 100

    For i As Integer = 1 To jumlahbaris
        drawline()
        y += 90
    Next

    x = 10
    y = 100
    cekxstart()
    cekystart(0)

    Using g As Graphics = Graphics.FromImage(bmp)
        g.DrawString(partarray(0).ToString, TextTitle, Brushes.Black,
10, 10)
        g.DrawString(partarray(1).ToString, TextSubTitle,
Brushes.Black, 10, 30)
        g.DrawString(partarray(2).ToString, TextSubTitle,
Brushes.Black, 10, 45)
        g.DrawString(partarray(3).ToString, TextSubTitle,
Brushes.Black, 10, 60)
    End Using

    Dim arraycount = UBound(partarray)

    For i As Integer = 4 To arraycount - 1 Step 1

        If partarray(i) <> "ENDPART" Then
            If partarray(i) <> "END" Then

                If partarray(i) = "B" Then
                    Using g As Graphics = Graphics.FromImage(bmp)
                        g.DrawLine(Pens.Black, xstart, y, xstart, y +
40)

                    End Using

```



```

xstart += 10

ElseIf partarray(i) <> "B" Then
    Dim temp As String = partarray(i).Substring(0, 1)

    Select Case temp
        Case "W"
            Dim t As String = partarray(i).Substring(1,
1)
            Dim tn As String =
partarray(i).Substring(2, 1)
            If tn = "0" Then
                Using g As Graphics =
Graphics.FromImage bmp)
                    g.DrawImage(imgwholerest, xstart, y
+ 10)
                End Using
            Else
                cekrangebalok(t, tn, imgwholenote)
            End If
            xstart += 20
        Case "H"
            Dim t As String = partarray(i).Substring(1,
1)
            Dim tn As String =
partarray(i).Substring(2, 1)
            If tn = "0" Then
                Using g As Graphics =
Graphics.FromImage bmp)
                    g.DrawImage(imghalfrest, xstart, y
+ 10)
                End Using
            Else
                cekrangebalok(t, tn, imghalfnote)
            End If
            xstart += 20
        Case "Q"
            Dim t As String = partarray(i).Substring(1,
1)
            Dim tn As String =
partarray(i).Substring(2, 1)
            If tn = "0" Then
                Using g As Graphics =
Graphics.FromImage bmp)
                    g.DrawImage(imgquaterrest, xstart,
y)
                End Using
            Else
                cekrangebalok(t, tn, imgquaternote)
            End If
            xstart += 20
        Case "E"
            Dim t As String = partarray(i).Substring(1,
1)
            Dim tn As String =
partarray(i).Substring(2, 1)
            Dim t1 As String =
partarray(i).Substring(4, 1)

```

```

partarray(i).Substring(5, 1)
Graphics.FromImage bmp)
y)
Dim tn1 As String =
If tn = "0" Then
    Using g As Graphics =
        g.DrawImage(imgeightrest, xstart,
            End Using
            xstart += 20
            cekrangebalok(t1, tn1, imgeightnote)
Else
    cekrangebalok(t, tn, imgquaternote)
    xstart += 20
    Using g As Graphics =
        g.DrawLine(Pens.Black, xstart - 5,
getyposition(t, tn) + 2, xstart + 15, getyposition(t1, tn1) + 2)
        End Using
        cekrangebalok(t1, tn1, imgquaternote)
    End If
    xstart += 20
Case "S"
Dim t As String = partarray(i).Substring(1,
1)
Dim tn As String =
partarray(i).Substring(2, 1)
Dim t1 As String =
partarray(i).Substring(4, 1)
Dim tn1 As String =
partarray(i).Substring(5, 1)
Dim t2 As String =
partarray(i).Substring(7, 1)
Dim tn2 As String =
partarray(i).Substring(8, 1)
Dim t3 As String =
partarray(i).Substring(10, 1)
Dim tn3 As String =
partarray(i).Substring(11, 1)
If tn = "0" Then
    Using g As Graphics =
        g.DrawImage(imgsixteenthrest,
xstart, y)
        End Using
        xstart += 20
        cekrangebalok(t1, tn1, imgeightnote)
        xstart += 20
        Using g As Graphics =
            g.DrawLine(Pens.Black, xstart - 5,
getyposition(t, tn) + 2, xstart + 15, getyposition(t1, tn1) + 2)
            g.DrawLine(Pens.Black, xstart - 4,
getyposition(t, tn) + 7, xstart + 15, getyposition(t1, tn1) + 7)
            End Using
            xstart += 20
            cekrangebalok(t2, tn2, imgeightnote)

```

```

xstart += 20
Using g As Graphics =
Graphics.FromImage bmp)
    g.DrawLine(Pens.Black, xstart - 5,
getyposition(t, tn) + 2, xstart + 15, getyposition(t1, tn1) + 2)
    g.DrawLine(Pens.Black, xstart - 4,
getyposition(t, tn) + 7, xstart + 15, getyposition(t1, tn1) + 7)
End Using
xstart += 20
cekrangebalok(t3, tn3, imgeightnote)

ElseIf tn1 = "0" Then
cekrangebalok(t, tn, imgsixteenthnote)
xstart += 20
Using g As Graphics =
Graphics.FromImage bmp)
    g.DrawImage(imgsixteenthrest,
xstart, y)
End Using
xstart += 20
cekrangebalok(t2, tn2, imgeightnote)
xstart += 20
Using g As Graphics =
Graphics.FromImage bmp)
    g.DrawLine(Pens.Black, xstart - 5,
getyposition(t, tn) + 2, xstart + 15, getyposition(t1, tn1) + 2)
    g.DrawLine(Pens.Black, xstart - 4,
getyposition(t, tn) + 7, xstart + 15, getyposition(t1, tn1) + 7)
End Using
xstart += 20
cekrangebalok(t3, tn3, imgeightnote)

ElseIf tn2 = "0" Then
cekrangebalok(t, tn, imgquaternote)
xstart += 20
Using g As Graphics =
Graphics.FromImage bmp)
    g.DrawLine(Pens.Black, xstart - 5,
getyposition(t, tn) + 2, xstart + 15, getyposition(t1, tn1) + 2)
    g.DrawLine(Pens.Black, xstart - 4,
getyposition(t, tn) + 7, xstart + 15, getyposition(t1, tn1) + 7)
End Using
cekrangebalok(t1, tn1, imgquaternote)
xstart += 20
Using g As Graphics =
Graphics.FromImage bmp)
    g.DrawImage(imgsixteenthrest,
xstart, y)
End Using
xstart += 20
cekrangebalok(t, tn, imgsixteenthnote)

ElseIf tn3 = "0" Then
cekrangebalok(t, tn, imgquaternote)
xstart += 20
If tn <> "0" Then
Using g As Graphics =
Graphics.FromImage bmp)
    g.DrawLine(Pens.Black, xstart -
5, getyposition(t, tn) + 2, xstart + 15, getyposition(t1, tn1) + 2)

```

```

        g.DrawLine(Pens.Black, xstart -
4, getyposition(t, tn) + 7, xstart + 15, getyposition(t1, tn1) + 7)
        End Using
    End If
    cekrangebalok(t1, tn1, imgquaternote)
    xstart += 20
    If tn <> "0" Then
        Using g As Graphics =
Graphics.FromImage bmp)
            g.DrawLine(Pens.Black, xstart -
5, getyposition(t, tn) + 2, xstart + 15, getyposition(t1, tn1) + 2)
            g.DrawLine(Pens.Black, xstart -
4, getyposition(t, tn) + 7, xstart + 15, getyposition(t1, tn1) + 7)
            End Using
        End If
        cekrangebalok(t2, tn2, imgquaternote)
        xstart += 20
        Using g As Graphics =
Graphics.FromImage bmp)
            g.DrawImage(imgsixteenthrest,
xstart, y)
            End Using
        End If
        xstart += 20
    End Select
End If

ElseIf partarray(i) = "END" Then
    cekxstart()
    tmpy += 90
    y += 90
    cekystart(tmpy)
End If
Else
    MessageBox.Show("Generate Done", "Sukses!")
End If
Next

frmNotBalok.picNotBalok.Image = bmp
End Sub

'program
Private Sub lsbFile_MouseUp(sender As Object, e As MouseEventArgs)
Handles lsbFile.MouseUp
    If FrmMain.bartxtStatus.Caption = "Done" Then
        Me.Cursor = New Cursor(Cursor.Current.Handle)
        Dim pt As Point = New Point(Cursor.Position.X - 50,
Cursor.Position.Y - 50)
        FrmMain.RadialMenu.ShowPopup(pt)
    End If
End Sub

Private Sub frmFile_FormClosing(sender As Object, e As
FormClosingEventArgs) Handles Me.FormClosing
    If vbOK = MessageBox.Show("Close file?", "Warning!",
MessageBoxButtons.OKCancel, MessageBoxIcon.Question) Then
        With FrmMain
            .rpgSound.Enabled = False
            .rpgGenerate.Enabled = False

```

```

        .bartxtStatus.Caption = "New Partitur"
        .initstatusbar()
    End With
    frmNotAngkaGenerate.Close()
    frmNotBalok.Close()
    frmNotAngka.Close()
Else
    e.Cancel = True
End If
End Sub

Private Sub frmFile_Shown(sender As Object, e As EventArgs) Handles
Me.Shown
    setarray()

    With FrmMain
        .bartxtTitle.Caption = "Title : " + partarray(0).ToString
        .bartxtAuthor.Caption = "Author : " + partarray(1).ToString
        .bartxtBar.Caption = partarray(2).ToString
        .bartxtScale.Caption = partarray(3).ToString
    End With

End Sub

'playmusic
Private Sub ceksound(range As String, note As String, dur As Int32)
    If range = "T" Then
        Select Case note
            Case "0"
                Thread.Sleep(dur)
            Case "1"
                Console.Beep(523, dur)
            Case "2"
                Console.Beep(587, dur)
            Case "3"
                Console.Beep(659, dur)
            Case "4"
                Console.Beep(698, dur)
            Case "5"
                Console.Beep(784, dur)
            Case "6"
                Console.Beep(880, dur)
            Case "7"
                Console.Beep(987, dur)
        End Select
    ElseIf range = "N" Then
        Select Case note
            Case "0"
                Thread.Sleep(dur)
            Case "1"
                Console.Beep(261, dur)
            Case "2"
                Console.Beep(293, dur)
            Case "3"
                Console.Beep(329, dur)
            Case "4"
                Console.Beep(349, dur)
            Case "5"
                Console.Beep(392, dur)
            Case "6"

```

```

        Console.Beep(440, dur)
    Case "7"
        Console.Beep(493, dur)
    End Select
ElseIf range = "R" Then
    Select Case note
    Case "0"
        Thread.Sleep(dur)
    Case "1"
        Console.Beep(130, dur)
    Case "2"
        Console.Beep(146, dur)
    Case "3"
        Console.Beep(164, dur)
    Case "4"
        Console.Beep(174, dur)
    Case "5"
        Console.Beep(196, dur)
    Case "6"
        Console.Beep(220, dur)
    Case "7"
        Console.Beep(246, dur)
    End Select
End If

End Sub

Public Sub playmusic(tmp As String)
    Dim bpm As Int32 = 0
    setarray()

    If tmp = "Adagio 60bpm" Then
        bpm = 1000
    ElseIf tmp = "Allegro 120bpm" Then
        bpm = 500
    End If

    Dim arraycount = UBound(partarray)

    For i As Integer = 4 To arraycount - 1 Step 1
        If partarray(i) <> "ENDPART" Then
            If partarray(i) <> "END" Then

                If partarray(i) = "B" Then
                    Thread.Sleep(250)
                ElseIf partarray(i) <> "B" Then
                    Dim temp As String = partarray(i).Substring(0, 1)
                    If temp = "W" Then
                        Dim t As String = partarray(i).Substring(1, 1)
                        Dim tn As String = partarray(i).Substring(2, 1)
                        ceksound(t, tn, bpm * 4)

                    ElseIf temp = "H" Then
                        Dim t As String = partarray(i).Substring(1, 1)
                        Dim tn As String = partarray(i).Substring(2, 1)
                        ceksound(t, tn, bpm * 2)

                    ElseIf temp = "Q" Then
                        Dim t As String = partarray(i).Substring(1, 1)
                        Dim tn As String = partarray(i).Substring(2, 1)

```

```

        ceksound(t, tn, bpm)

    ElseIf temp = "E" Then
        Dim t As String = partarray(i).Substring(1, 1)
        Dim tn As String = partarray(i).Substring(2, 1)

        Dim t1 As String = partarray(i).Substring(4, 1)
        Dim tn1 As String = partarray(i).Substring(5, 1)
1)        ceksound(t, tn, Convert.ToInt32(bpm / 2))

        ceksound(t1, tn1, Convert.ToInt32(bpm / 2))
    ElseIf temp = "S" Then
        Dim t As String = partarray(i).Substring(1, 1)
        Dim tn As String = partarray(i).Substring(2, 1)

        Dim t1 As String = partarray(i).Substring(4, 1)
        Dim tn1 As String = partarray(i).Substring(5, 1)
1)

        Dim t2 As String = partarray(i).Substring(7, 1)
        Dim tn2 As String = partarray(i).Substring(8, 1)
1)

        Dim t3 As String = partarray(i).Substring(10, 1)
        Dim tn3 As String = partarray(i).Substring(11, 1)
1)
1)        ceksound(t, tn, Convert.ToInt32(bpm / 4))

        ceksound(t1, tn1, Convert.ToInt32(bpm / 4))

        ceksound(t2, tn2, Convert.ToInt32(bpm / 4))

        ceksound(t3, tn3, Convert.ToInt32(bpm / 4))
    End If
    End If
    ElseIf partarray(i) = "END" Then
    End If
    Else
        MessageBox.Show("Play Done", "Sukses!")
    End If
    Next
End Sub

End Class

```

```

Imports System.ComponentModel
Imports System.Text
Imports System.Drawing.Graphics

Public Class frmNotAngka
    Dim x, y, tmpnot, tmpbar, tmpketuk As Integer
    Dim jmlnot, jmlbar, jmlketuk As Double
    Dim codenote, tempfile As String
    Dim TextTitle As New System.Drawing.Font("calibri", 4, FontStyle.Bold)
    Dim TextSubTitle As New System.Drawing.Font("calibri", 3,
FontStyle.Bold)
    Dim TextNot As New System.Drawing.Font("calibri", 3, FontStyle.Regular)
    Dim TextDot As New System.Drawing.Font("calibri", 4, FontStyle.Bold)
    Dim bmp As Image = My.Resources.LayerNot

    Private m_PanStartPoint As New Point

    Protected Overrides Sub DefWndProc(ByRef m As Message)
        If m.Msg <> 131 Then
            MyBase.DefWndProc(m)
        End If
    End Sub

    Private Sub disableallfield()
        groupNotation.Enabled = False
        groupNotes.Enabled = False
        groupSign.Enabled = False
        btnSubmit.Enabled = False
        btnDone.Enabled = False
    End Sub

    'draw not angka
    Public Sub cekrange(tmprange As String, tmpnot As String, tmpsgn As
String)
        If tmprange = "T" Then
            If tmpsgn = "N" Then
                Using g As Graphics = Graphics.FromImage(bmp)
                    g.DrawString(tmpnot, TextNot, Brushes.Black, x, y)
                    g.DrawString(".", TextDot, Brushes.Black, x + 1, y -
13)
                End Using
            Else
                Using g As Graphics = Graphics.FromImage(bmp)
                    g.DrawString(tmpnot, TextNot, Brushes.Black, x, y)
                    g.DrawString(".", TextDot, Brushes.Black, x + 1, y -
13)
                    g.DrawLine(Pens.Black, x, y + 10, x + 7, y + 3)
                End Using
            End If
        ElseIf tmprange = "R" Then
            If tmpsgn = "N" Then
                Using g As Graphics = Graphics.FromImage(bmp)
                    g.DrawString(tmpnot, TextNot, Brushes.Black, x, y)
                    g.DrawString(".", TextDot, Brushes.Black, +1, y + 13)
                End Using
            Else
                Using g As Graphics = Graphics.FromImage(bmp)
                    g.DrawString(tmpnot, TextNot, Brushes.Black, x, y)
                    g.DrawString(".", TextDot, Brushes.Black, +1, y + 13)
                End Using
            End If
        End Sub

```



```

        g.DrawLine(Pens.Black, x, y + 10, x + 7, y + 3)
    End Using
End If

ElseIf tmprange = "N" Then
    If tmpsgn = "N" Then
        Using g As Graphics = Graphics.FromImage(bmp)
            g.DrawString(tmpnot, TextNot, Brushes.Black, x, y)
        End Using
    Else
        Using g As Graphics = Graphics.FromImage(bmp)
            g.DrawString(tmpnot, TextNot, Brushes.Black, x, y)
            g.DrawLine(Pens.Black, x, y + 10, x + 7, y + 3)
        End Using
    End If
End If
End Sub

Public Sub drawnotangka()
    If codenote <> "END" Then

        If codenote = "B" Then
            Using g As Graphics = Graphics.FromImage(bmp)
                g.DrawLine(Pens.Black, x, y + 10, x, y - 10)
            End Using
            x += 20

        ElseIf codenote <> "B" Then
            Dim temp As String = codenote.Substring(0, 1)

            If temp = "W" Then
                Dim t As String = codenote.Substring(1, 1)
                Dim tn As String = codenote.Substring(2, 1)
                Dim ts As String = codenote.Substring(3, 1)
                cekrange(t, tn, ts)
                Using g As Graphics = Graphics.FromImage(bmp)
                    g.DrawString(".", TextDot, Brushes.Black, x + 20,
y)
                    g.DrawString(".", TextDot, Brushes.Black, x + 40,
y)
                    g.DrawString(".", TextDot, Brushes.Black, x + 60,
y)
                End Using
                x += 80
            ElseIf temp = "H" Then
                Dim t As String = codenote.Substring(1, 1)
                Dim tn As String = codenote.Substring(2, 1)
                Dim ts As String = codenote.Substring(3, 1)
                cekrange(t, tn, ts)
                Using g As Graphics = Graphics.FromImage(bmp)
                    g.DrawString(".", TextDot, Brushes.Black, x + 20,
y)
                End Using
                x += 40
            ElseIf temp = "Q" Then
                Dim t As String = codenote.Substring(1, 1)
                Dim tn As String = codenote.Substring(2, 1)
                Dim ts As String = codenote.Substring(3, 1)
                cekrange(t, tn, ts)
                x += 20
            End If
        End If
    End If
End Sub

```

```

ElseIf temp = "E" Then
    Dim t As String = codenote.Substring(1, 1)
    Dim tn As String = codenote.Substring(2, 1)
    Dim ts As String = codenote.Substring(3, 1)
    Dim t1 As String = codenote.Substring(4, 1)
    Dim tn1 As String = codenote.Substring(5, 1)
    Dim ts1 As String = codenote.Substring(6, 1)
    cekrange(t, tn, ts)
    x += 20
    cekrange(t1, tn1, ts1)
    Using g As Graphics = Graphics.FromImage bmp
        g.DrawLine(Pens.Black, x - 20, y - 10, x + 5, y -
10)

    End Using
    x += 20
ElseIf temp = "S" Then
    Dim t As String = codenote.Substring(1, 1)
    Dim tn As String = codenote.Substring(2, 1)
    Dim ts As String = codenote.Substring(3, 1)
    Dim t1 As String = codenote.Substring(4, 1)
    Dim tn1 As String = codenote.Substring(5, 1)
    Dim ts1 As String = codenote.Substring(6, 1)
    Dim t2 As String = codenote.Substring(7, 1)
    Dim tn2 As String = codenote.Substring(8, 1)
    Dim ts2 As String = codenote.Substring(9, 1)
    Dim t3 As String = codenote.Substring(10, 1)
    Dim tn3 As String = codenote.Substring(11, 1)
    Dim ts3 As String = codenote.Substring(12, 1)
    cekrange(t, tn, ts)
    x += 20
    cekrange(t1, tn1, ts1)
    x += 20
    cekrange(t2, tn2, ts2)
    x += 20
    cekrange(t3, tn3, ts3)
    Using g As Graphics = Graphics.FromImage bmp
        g.DrawLine(Pens.Black, x - 60, y - 10, x + 5, y -
10)

    End Using
    x += 20
End If
End If
ElseIf codenote = "END" Then
    x = 10
    y += 35
End If

picNotAngka.Image = bmp
End Sub

'cek not and birama
Public Sub cekjumlah()
    If tmpnot = jmlnot Then
        groupNotes.Enabled = False
        groupNotation.Enabled = False
        groupSign.Enabled = False
        btnSubmit.Enabled = True
        jmlnot = 0
    Else
        groupNotes.Enabled = False

```

```

        groupNotation.Enabled = True
        groupSign.Enabled = False
        btnSubmit.Enabled = False
    End If
End Sub

'cek jumlah bar
Public Sub cekbar()
    If FrmMain.bartxtBar.Caption = "3/4" Then
        jmlbar = 4
        jmlketuk = 3
        btnFull.Enabled = False
    ElseIf FrmMain.bartxtBar.Caption = "4/4" Then
        jmlbar = 4
        jmlketuk = 4
        btnFull.Enabled = True
    ElseIf FrmMain.bartxtBar.Caption = "6/8" Then
        jmlbar = 8
        jmlketuk = 6
        btnFull.Enabled = True
    End If
End Sub

#Region "Notes"
Private Sub btnFull_Click(sender As Object, e As EventArgs) Handles
btnFull.Click
    txtInput.Text = ""
    txtInput.Text += btnFull.Tag.ToString
    tmpnot = 1
    tmpketuk += 4
    groupNotes.Enabled = False
    groupNotation.Enabled = True
End Sub

Private Sub btnHalf_Click(sender As Object, e As EventArgs) Handles
btnHalf.Click
    txtInput.Text = ""
    txtInput.Text += btnHalf.Tag.ToString
    tmpnot = 1
    tmpketuk += 2
    groupNotes.Enabled = False
    groupNotation.Enabled = True
End Sub

Private Sub btnQuarter_Click(sender As Object, e As EventArgs) Handles
btnQuarter.Click
    txtInput.Text = ""
    txtInput.Text += btnQuarter.Tag.ToString
    tmpnot = 1
    tmpketuk += 1
    groupNotes.Enabled = False
    groupNotation.Enabled = True
End Sub

Private Sub btnEighth_Click(sender As Object, e As EventArgs) Handles
btnEighth.Click
    txtInput.Text = ""
    txtInput.Text += btnEighth.Tag.ToString
    tmpnot = 2
    tmpketuk += 1

```

```

        groupNotes.Enabled = False
        groupNotation.Enabled = True
    End Sub

    Private Sub btnSixteenth_Click(sender As Object, e As EventArgs)
Handles btnSixteenth.Click
        txtInput.Text = ""
        txtInput.Text += btnSixteenth.Tag.ToString
        tmpnot = 4
        tmpketuk += 1
        groupNotes.Enabled = False
        groupNotation.Enabled = True
    End Sub
#End Region

#Region "Notation"
    Private Sub btnH1_Click(sender As Object, e As EventArgs) Handles
btnH1.Click
        txtInput.Text += btnH1.Tag.ToString
        groupNotation.Enabled = False
        groupSign.Enabled = True
    End Sub

    Private Sub btnH2_Click(sender As Object, e As EventArgs) Handles
btnH2.Click
        txtInput.Text += btnH2.Tag.ToString
        groupNotation.Enabled = False
        groupSign.Enabled = True
    End Sub

    Private Sub btnH3_Click(sender As Object, e As EventArgs) Handles
btnH3.Click
        txtInput.Text += btnH3.Tag.ToString
        groupNotation.Enabled = False
        groupSign.Enabled = True
    End Sub

    Private Sub btnH4_Click(sender As Object, e As EventArgs) Handles
btnH4.Click
        txtInput.Text += btnH4.Tag.ToString
        groupNotation.Enabled = False
        groupSign.Enabled = True
    End Sub

    Private Sub btnH5_Click(sender As Object, e As EventArgs) Handles
btnH5.Click
        txtInput.Text += btnH5.Tag.ToString
        groupNotation.Enabled = False
        groupSign.Enabled = True
    End Sub

    Private Sub btnH6_Click(sender As Object, e As EventArgs) Handles
btnH6.Click
        txtInput.Text += btnH6.Tag.ToString
        groupNotation.Enabled = False
        groupSign.Enabled = True
    End Sub

    Private Sub btnH7_Click(sender As Object, e As EventArgs) Handles
btnH7.Click

```

```
        txtInput.Text += btnH7.Tag.ToString
        groupNotation.Enabled = False
        groupSign.Enabled = True
    End Sub

    Private Sub btnN1_Click(sender As Object, e As EventArgs) Handles
btnN1.Click
        txtInput.Text += btnN1.Tag.ToString
        groupNotation.Enabled = False
        groupSign.Enabled = True
    End Sub

    Private Sub btnN2_Click(sender As Object, e As EventArgs) Handles
btnN2.Click
        txtInput.Text += btnN2.Tag.ToString
        groupNotation.Enabled = False
        groupSign.Enabled = True
    End Sub

    Private Sub btnN3_Click(sender As Object, e As EventArgs) Handles
btnN3.Click
        txtInput.Text += btnN3.Tag.ToString
        groupNotation.Enabled = False
        groupSign.Enabled = True
    End Sub

    Private Sub btnN4_Click(sender As Object, e As EventArgs) Handles
btnN4.Click
        txtInput.Text += btnN4.Tag.ToString
        groupNotation.Enabled = False
        groupSign.Enabled = True
    End Sub

    Private Sub btnN5_Click(sender As Object, e As EventArgs) Handles
btnN5.Click
        txtInput.Text += btnN5.Tag.ToString
        groupNotation.Enabled = False
        groupSign.Enabled = True
    End Sub

    Private Sub btnN6_Click(sender As Object, e As EventArgs) Handles
btnN6.Click
        txtInput.Text += btnN6.Tag.ToString
        groupNotation.Enabled = False
        groupSign.Enabled = True
    End Sub

    Private Sub btnN7_Click(sender As Object, e As EventArgs) Handles
btnN7.Click
        txtInput.Text += btnN7.Tag.ToString
        groupNotation.Enabled = False
        groupSign.Enabled = True
    End Sub

    Private Sub btnR1_Click(sender As Object, e As EventArgs) Handles
btnR1.Click
        txtInput.Text += btnR1.Tag.ToString
        groupNotation.Enabled = False
        groupSign.Enabled = True
    End Sub
```

```

    Private Sub btnR2_Click(sender As Object, e As EventArgs) Handles
btnR2.Click
        txtInput.Text += btnR2.Tag.ToString
        groupNotation.Enabled = False
        groupSign.Enabled = True
    End Sub

    Private Sub btnR3_Click(sender As Object, e As EventArgs) Handles
btnR3.Click
        txtInput.Text += btnR3.Tag.ToString
        groupNotation.Enabled = False
        groupSign.Enabled = True
    End Sub

    Private Sub btnR4_Click(sender As Object, e As EventArgs) Handles
btnR4.Click
        txtInput.Text += btnR4.Tag.ToString
        groupNotation.Enabled = False
        groupSign.Enabled = True
    End Sub

    Private Sub btnR5_Click(sender As Object, e As EventArgs) Handles
btnR5.Click
        txtInput.Text += btnR5.Tag.ToString
        groupNotation.Enabled = False
        groupSign.Enabled = True
    End Sub

    Private Sub btnR6_Click(sender As Object, e As EventArgs) Handles
btnR6.Click
        txtInput.Text += btnR6.Tag.ToString
        groupNotation.Enabled = False
        groupSign.Enabled = True
    End Sub

    Private Sub btnR7_Click(sender As Object, e As EventArgs) Handles
btnR7.Click
        txtInput.Text += btnR7.Tag.ToString
        groupNotation.Enabled = False
        groupSign.Enabled = True
    End Sub

    Private Sub btnRest_Click(sender As Object, e As EventArgs) Handles
btnRest.Click
        txtInput.Text += btnRest.Tag.ToString
        groupNotation.Enabled = False
        groupSign.Enabled = True
        btnCress.Enabled = False
    End Sub
#End Region

#Region "Sign"
    Private Sub btnCress_Click(sender As Object, e As EventArgs) Handles
btnCress.Click
        txtInput.Text += btnCress.Tag.ToString
        jmlnot += 1
        cekjumlah()
    End Sub

```

```

    Private Sub btnNoSign_Click(sender As Object, e As EventArgs) Handles
btnNoSign.Click
        txtInput.Text += btnNoSign.Tag.ToString
        jmlnot += 1
        cekjumlah()
        btnCress.Enabled = True
    End Sub
#End Region

```

```

    Private Sub btnDone_Click(sender As Object, e As EventArgs) Handles
btnDone.Click
        Dim i As Integer = 0
        If vbOK = MessageBox.Show("Done?", "Warning!",
MessageBoxButtons.OKCancel, MessageBoxIcon.Question) Then
            FrmMain.bartxtStatus.Caption = "Done"
            FrmMain.cekstatus()
            i = frmFile.lsbFile.Items.Count - 1
            If frmFile.lsbFile.Items(i) Is "End" Then
                frmFile.lsbFile.Items.Add("ENDPART:")
            Else
                frmFile.lsbFile.Items.Add("END:")
                frmFile.lsbFile.Items.Add("ENDPART:")
            End If
            disnableallfield()
        End If

    End Sub

```

```

    Private Sub btnSubmit_Click(sender As Object, e As EventArgs) Handles
btnSubmit.Click
        codenote = txtInput.Text
        drawnotangka()

        If tmpketuk >= jmlketuk Then
            tmpketuk = 0
            tempfile += txtInput.Text + ":B:"
            frmFile.lsbFile.Items.Add(tempfile)
            tempfile = ""
            tmpbar += 1
            codenote = "B"
            drawnotangka()
        ElseIf tmpketuk < jmlketuk Then
            tempfile += txtInput.Text + ":"
        End If

        If tmpbar = jmlbar Then
            frmFile.lsbFile.Items.Add("END:")
            tmpbar = 0
            x = 10
            y += 35
        End If

        txtInput.Text = ""
        groupNotes.Enabled = True
        btnSubmit.Enabled = False
    End Sub

```

```

    Private Sub frmNotAngka_Load(sender As Object, e As EventArgs) Handles
MyBase.Load
        Panel1.AutoScroll = True

```

```

picNotAngka.SizeMode = PictureBoxSizeMode.AutoSize
x = 10
y = 85
btnSubmit.Enabled = False
bmp = picNotAngka.Image
tempfile = ""
groupNotes.Enabled = True
groupNotation.Enabled = False
groupSign.Enabled = False
cekbar()
End Sub

Private Sub picNotAngka_MouseDown(sender As Object, e As
MouseEventArgs) Handles picNotAngka.MouseDown
    m_PanStartPoint = New Point(e.X, e.Y)
End Sub

Private Sub picNotAngka_MouseMove(sender As Object, e As
MouseEventArgs) Handles picNotAngka.MouseMove
    If e.Button = Windows.Forms.MouseButtons.Left Then

        Dim DeltaX As Integer = (m_PanStartPoint.X - e.X)
        Dim DeltaY As Integer = (m_PanStartPoint.Y - e.Y)

        Panel1.AutoScrollPosition = _
            New Drawing.Point((DeltaX - Panel1.AutoScrollPosition.X), _
                (DeltaY - Panel1.AutoScrollPosition.Y))
    End If
End Sub
End Class

```



```

Public Class frmNotAngkaGenerate

    Private m_PanStartPoint As New Point

    Protected Overrides Sub DefWndProc(ByRef m As Message)
        If m.Msg <> 131 Then
            MyBase.DefWndProc(m)
        End If
    End Sub

    Private Sub frmNotAngkaGenerate_FormClosed(sender As Object, e As
    FormClosedEventArgs) Handles MyBase.FormClosed
        FrmMain.btnExportNotAngka.Enabled = False
        Panel1.AutoScroll = True
        picAngkaGenerate.SizeMode = PictureBoxSizeMode.AutoSize
    End Sub

    Private Sub frmNotAngkaGenerate_Load(sender As Object, e As EventArgs)
    Handles MyBase.Load
        FrmMain.btnExportNotAngka.Enabled = True
    End Sub

    Private Sub picAngkaGenerate_MouseDown(sender As Object, e As
    MouseEventArgs) Handles picAngkaGenerate.MouseDown
        m_PanStartPoint = New Point(e.X, e.Y)
    End Sub

    Private Sub picAngkaGenerate_MouseMove(sender As Object, e As
    MouseEventArgs) Handles picAngkaGenerate.MouseMove
        If e.Button = Windows.Forms.MouseButtons.Left Then

            Dim DeltaX As Integer = (m_PanStartPoint.X - e.X)
            Dim DeltaY As Integer = (m_PanStartPoint.Y - e.Y)

            Panel1.AutoScrollPosition = _
            New Drawing.Point((DeltaX - Panel1.AutoScrollPosition.X), _
            (DeltaY - Panel1.AutoScrollPosition.Y))

        End If
    End Sub

End Class

```

```

Public Class frmNotBalok

    Private m_PanStartPoint As New Point

    Protected Overrides Sub DefWndProc(ByRef m As Message)
        If m.Msg <> 131 Then
            MyBase.DefWndProc(m)
        End If
    End Sub

    Private Sub frmNotBalok_Load(sender As Object, e As EventArgs) Handles
MyBase.Load
        FrmMain.btnExportNotBalok.Enabled = True
        Panel1.AutoScroll = True
        picNotBalok.SizeMode = PictureBoxSizeMode.AutoSize
    End Sub

    Private Sub frmNotBalok_FormClosed(sender As Object, e As
FormClosedEventArgs) Handles MyBase.FormClosed
        FrmMain.btnExportNotBalok.Enabled = False
    End Sub

    Private Sub picNotBalok_MouseDown(sender As Object, e As
MouseEventArgs) Handles picNotBalok.MouseDown
        m_PanStartPoint = New Point(e.X, e.Y)
    End Sub

    Private Sub picNotBalok_MouseMove(sender As Object, e As
MouseEventArgs) Handles picNotBalok.MouseMove
        If e.Button = Windows.Forms.MouseButtons.Left Then

            Dim DeltaX As Integer = (m_PanStartPoint.X - e.X)
            Dim DeltaY As Integer = (m_PanStartPoint.Y - e.Y)

            Panel1.AutoScrollPosition = _
                New Drawing.Point((DeltaX - Panel1.AutoScrollPosition.X), _
                    (DeltaY - Panel1.AutoScrollPosition.Y))
        End If
    End Sub
End Class

```