BAB II

TINJAUAN PUSTAKA

II.1. Perancangan

Perancangan adalah kegiatan awal dari suatu rangkaian kegiatan dalam proses pembuatan produk. Dalam tahap perancangan tersebut dibuat keputusan-keputusan penting yang mempengaruhi kegiatan-kegiatan lain yang menyusulnya. Perancangan yang dimaksud adalah sebuah proses membuat beberapa *output* media komunikasi *visual* yang didasari adanya sebuah kebutuhan atau suatu permasalahan, mulai dari perencanaan, pengumpulan dan analisa data hingga membuat desain yang *efisien* dan sesuai tujuan.

Demi mencapai prestasi belajar yang memuaskan tersebut dengan sistem pendidikan perkuliahan yang semakin maju dan didukung juga perkembangan teknologi. Teknologi multimedia telah menjanjikan potensi besar dalam merubah cara seseorang untuk belajar, untuk memperoleh informasi, menyesuaikan informasi dan sebagainya. (Galih pranowo:2010;1).

II.1.1. Multimedia

Multi-banyak, Media sarana berkomunikasi untuk melewat informasi. Suatu sitem yang terdiri dari perangkat keras, perangkat lunak dan alat lain seperti televisi, monitor video dan sistem piringan optik atau sistem stero yang dimaksudkan untuk menghasilkan penyajian audio visual yang utuh.

Beberapa Pakar mengartikan multimedia sebagai berikut:

1. Multimedia secara umum merupakan kombinasi 3 elemen yaitu suara, gambar dan teks.
2. Multimedia adalah kombinasi dari paling sedikit 2 media input dan output dari data, media ini dapat audio(suara, musik), animasi, video, teks, grafik, dan gambar.
3. Multimedia merupakan alat yang dapat menciptakan prestasi yang dinamis dan interaktif yang mengkombinasi teks grafik, animasi,audio dan gambar video.
4. Multimedia adalah pemanfaatan komputer untuk membuat dan menggabungkan teks,grafik,audio,gambar bergerak(video,animasi) dengan menggabung link dan tool yang memungkinkan pemakai malakukan navigasi,berintraksi,berkreasi dan berkomunikasi. (Chrisna Atmadji;2010).

II.1.2. Siklus Hidup Pengembangan Sistem

Metode perangkat lunak yang digunakan dalam penelitian adalah dengan pendekatan siklus hidup pengembangan system yang selanjutnya disebut SHPS. SHPS adalah pendekatan melalui beberapa tahap untuk menganalisis dan merancang sistem yang dimana sistem tersebut telah dikembangkan dengan baik melalui penggunaan siklus penganalisisdan pemakai secara spesific.

Tahap-tahap SHPS adalah sebagai berikut:

1. Mengindentifikasi masalah, peluang dan tujuan
2. Menentukan Syarat-syarat
3. Menganalisis kebutuhan sistem
4. Merancang sistem
5. Mengembangkan dan mendokumentasiskan sistem
6. Menguji dan mempertahankan sistem

7. Mengimplemetasikan dan mengepaluasi

Tahap ini adalah tahap perancangan suatu sistem untuk nantinya diimplementasikan menjadi sebuar perangkat lunak. Pada pengajaran Kinimatika Gerak Lurus, Gerak Melingkar dan Gerak Parabola berbasi komputer menggunakan *Data Flow Diagram (DFD)* sebagai pemodelan sistem yang telah dijabarkan sebelumnya. Pada taham ini akan dilalui taham merancang sistem yang direkomendasikan sebelumnya sesuai dengan siklus hidup pengembang sistem (SHPS). Dalam tahap desain dari (SHPS). Penulis menggunakan informasi yang telah terkumpul sebelumnya untuk mencapai desain sistem informasi yang logik. Penulis merancang prosedur data-entry sedemikian rupa sehingga data yang dimasukkan kedalam sistem benar benar akurat selain itu penulis menggunakan tehnik-tehnik bentuk perancangan layar tertentu untuk hasil yang lebih baik. (Novri Hadinata;2010).

II.1.3. Script dasar anamasi dan Basis-basis lainnya

Pada dasarnya Actionscript 3 memiliki struktur yang hampir sama seperti bahasa pemrograman pada umumnya sehingga apabila anda pernah belajar bahasa pemrograman seperti C, Delphi, javascript, atau yang lain, anda akan menemukan beberapa kesamaan yang akan mempermudah anda dalam memahami kode Actionscript.

1. Membuat dan menggunakan variabel

   Variabel dapat didefenisikan sebagai suatu nilai. Variabel memiliki berbagai tipe di antaranya adalah Number, String, Boolean, Array, dan sebagainya. Pendeklarisian variabel dapat dilakukan dengan cara menuliskan nama variabel diikuti dengan tipe variabel dan nilai variabel. Berikut in contoh pendeklarasian variabel.

Private var carSpeed:Number =10;.

2. Operasi kondisi(Conditional statement)

   Pernyataan kondisi *if* sering kali digunakan sebagai dasar logika dari sebuah pemrograman. Dalam Actionscript 3, penulisan *if* hampir sama dengan bahasa pemrograman lainnnya. Contoh :

   ```
   if (scorePemain == 3){
    menang();
    }
   ```

3. Operasi berulang(loop statement)

   Operasi berulang atau frloop adalah sebuah prosedur standar di mana perintah yang berada di dalam blok loop akan dijalankan secara berulang atau terus-menerus sampai kondisi tertentu terpenuhi. Actionscript 3 memiliki dua bentuk operasi berulang, yaitu *for* dan *while*. Perhatikan contoh kode menggunakan operasi *for* berikut:

   ```
   var score:Number = 0;
   for (var i:Number = 0; i<10; i++){
   score++;
   trace(score);
   }
   ```

4. Fungsi

   Fungsi adalah kumpulan dari satu atau beberapa perintah yang dapat digunakan secara berulang. Terdapat 4 elemen utama dari sebuah fungsi yaitu: deklarasi dari fungsi, parameter yang digunakan, output atau nilai yang dihasilkan oleh fungsi, dan kode pembentuk fungsi. Perhatikan contoh berikut:

   ```
   private function kuadrat(var num:Number):Number{
   ```

```
return num*num;
}
```

5. Event

   Event merupakan sebuah runtutan perintah yang dijalankan oleh objek tertentu ketika suatu “event” terjadi/berlangsung. Contoh dari “event”diantaranya adalah MouseEvent yaitu ketika input diperoleh dari mouse (*click, doubleClick, rollOver, rollOut,* dan sebagainya), keyboard Event(*keyUp, keyDown*, dan sebagainya), Timeline Event(*enterFrame* dan sebagainya), dan event lainnya. Untuk menambahkan sebuah event kepada objek dapat dilakukan dengan kode addEventListener. Perhatikan contoh berikut:

```
Bola_mc.addEventListener(Event.ENTER_FRAME, gerakanBola);
function gerakanBola(e:Event){
 var ob:Object = e.currentTarget;
ob.x += 5;
}
```

6. Class

   Dalam sebuah pemrograman berbasis objek atau OOP, pemahaman terhadap konsep sebuah objek sangatlah penting. Untuk mendapatkan gambaran sederhana tentang OOP, perhatikan contoh berikut: sebuah sepeda motor terbentuk dari mesin, kerangka, roda, lampu, dan sebagainya. Masing-masing elemen tersebut merupakan sebuah objek. Masing-masing objek memiliki fungsi spesifik, namun tidak akan berfungsi seutuhnya jika tidak digabung dengan objek lain. Mesin dapat berfungsi atau menyala, namun membutuhkan kerangkan, roda, dan elemen lain agar bisa disebut sebgai sepeda motor. Ketika semua objek bergabung dan

menjalankan fungsinya masing-masing, maka jadilah sebuah sepeda motor yang dapat berfungsi secara utuh.

Pembagian proses kerja menjadi beberapa objek spesifik akan mendapatkan beberapa keuntungan, antara lain:

1. ***Maintenance.*** Apabila terjadi sebuah kesalahan, kita tidak perlu membongkar semua sepeda motor tersebut, namun cukup memperbaiki bagian yang tidak bekerja dengan baik
2. ***Specialization.*** Dengan memisahkan objek secara spesifik, maka masing-masing objek akan bekerja sesuai dengan keahliannya.
3. ***Fault isolation*****.** Dengan memisahkan kerja masing-masing objek, kita dapat mengisolasi dan mendeteksi kesalahan yang muncul.(Wanda Wibawanto;2013).

II.2. Animasi

II.2.1. Sejarah Animasi

Sejak jaman purbakala manusia sudah memiliki bakat dalam membuat sebuah gambar, ini dibuktikan berdasarkan banyaknya ditemukan gambar-gambar yang terdapat di gua-gua purbakala atau bangunan-bangunan peninggalan jaman purbakala.

Gambar-gambar yang ada dianggap sebagai rekaman kejadian yang terjadi di masa itu, di abadikan dengan gambar-gambar bersambung sehingga menjadi sebuah cerita tersendiri yang dapat di mengerti oleh manusia jaman sekarang yang tentunya dengan pendekatan-pendekatan ilmu pengetahuan sekarang. Meskipun arah dari kejadian tersebut merupakan bagian penelusuran dari sejarah, namun dapat pula menjadi sebuah pedoman bahwa manusia memiliki kemampuan menggambar dan membuat cerita dari gambar-gambar yang di lukis. Di jaman sekarang, cerita dari gambar dapat di identikkan dengan komik atau cergam (cerita bergambar).

Komik atau cerita bergambar merupakan gabungan dari seni gambar dan kemampuan seseorang atau kelompok orang dalam membuat cerita. Dengan adanya sebuah alur cerita yang cukup panjang dan visualisasi cerita, ekspresi, dan karakter, cerita tersebut akan lebih menarik. Gambar-gambar tersebut berupa potongan-potongan kejadian cerita, meskipun tidak seperti visualisasi sebuah film yang terlihat nyata karena film adalah rekaman gambar bergerak, namun pembaca dapat berimajinasi dan mengerti alur dari cerita tersebut. Akan tetapi dari cerita gambar pun dapat dijadikan sebagai bahan untuk membuat sebuah film animasi. Perkembangan film animasi di negara barat sangat pesat, hal ini ditandai dengan banyaknya film-film animasi yang semakin berkembang dari tahun ke tahun. Seperti film petualangan *Micky Mouse* yang di produksi *Walt Disney* sampai dengan film kepahlawanan Superman dan Batman yang di produksi oleh *Warner Brothers*, hingga film-fim animasi yang banyak mengadopsi karakter hewan dari *Walt Disney* dengan film *Animaniac* dan *Looney Toon*.

II.2.2. Pengertian Animasi

Animasi berasal dari kata *animation* atau *to animate* yang berarti menghidupkan dalam kamus bahasa inggris-indonesia. Dalam kaitannya dengan cerita dan gambar, penulis beranggapan bahwa animasi merupakan kegiatan menghidupkan sebuah cerita dari beberapa gambar yang berkesinambungan sehingga ketika dalam proses animasi gambar tersebut terlihat seperti hidup. Secara umum animasi adalah kegiatan menghidupkan gambar mati agar terlihat hidup dan memiliki jiwa mirip dengan aslinya.

Animasi adalah gambar bergerak yang terbentuk dari sekumpulan objek (gambar) yang disusun secara beraturan mengikuti alur pergerakan yang telah

ditentukan pada setiap pertambahan hitungan waktu yang terjadi. Gambar atau objek yang dimaksud dalam defenisi diatas bisa berupa gambar manusia, hewan, maupun tulisan.

II.2.3. Jenis-jenis Animasi

*S*ecara umum animasi dibagi menjadi sembilan macam, yaitu :

1. Animasi Sell (*Cell Animation*), yaitu *Celluloid* yang merupakan material yang digunakan untuk membuat film gambar bergerak pada saat awal.
2. Animasi Frame (*Frame Animation*) yang merupakan bentuk animasi paling sederhana.
3. Animasi *Sprite (Sprite Animation)* adalah setiap bagian dari animasi anda yang bergerak secara mandiri, misalnya burung terbang, planet berotasi, bola memantul-mantul atau logo berputar.
4. Animasi Lintasan (*Path Animation*), yaitu animasi dari objek yang bergerak sepanjang garis kurva yang anda tentukan sebagai lintasan.
5. Animasi *Spline* adalah representasi dari matematis dari kurva. Bila objek bergerak, biasanya tidak mengikuti garis lurus, misalnya berbentuk kurva.
6. Animasi Vektor (*Vector Animation*), dimana sebuah vector merupakan garis yang memiliki ujung-pangkal, arah, dan panjang. Animasi vektor serupa dengan animasi sprite. Animasi sprite mengunakan bitmap untuk sprite, animasi vector menggunakan rumus matematika untuk menggambarkan sprite.
7. Animasi Karakter (*Character Animation*) merupakan sebuah cabang khusus animasi. Animasi karakter semacam yang anda lihat film kartun. Animasi ini berbeda dengan animasi lainnya, misalnya grafik bergerak animasi logo yang

melibatkan bentuk organic yang komplek dengan penggandaan yang banyak, gerakan yang hirarkis.

8. *Computational Animation,* menggerakkan objek di layar anda cukup memvariasikan koordinat x dan y-nya. Koordinat x merupakan posisi horizontal objek, yaitu berapa jauh kiri-kanan layar. Koordinat y merupakan posisi vertikal, yakni berapa jauh atas-bawah layar.
9. *Morphing* artinya mengubah satu bentuk lain dengan menampilkan serangkaian frame yang menciptakan gerakan halus begitu bentuk pertama mengubah dirinya menjadi bentuk lain.

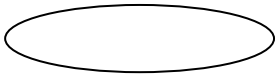
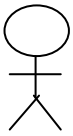
II.3. *Use Case*

Use Case adalah deskripsi fungsi bdari sebuah sistem dari *perspektif* pengguna. *Use case* bekerja dengan cara mendeeskripsikan tipikal intraksi antara user (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagamana sebuah sistem dipakai. Model use case adalah bagian dari *requirement. Use case* adalah alat bantu terbaik guna menstimulasi pengguna potensial untuk mengatakan tentang suatu sistem dari sudut pandangnya. Dengan demikian diharapkan akan bisa dibangun suatu sistem yang bisa membantu pengguna, perlu diingat bahwa *use case* mewakili pandangan diluar sistem. Diagram *Use case* menunjukkan 3 aspek dari sistem yaitu: actor, *use case dan system / sub systemboundary. Actor* mewakili peran orang, *system* yang lain atau ketika berkomunikasi dengan use case. Berikut gambar notasi use case:

Berikut ini gambar table simbol-simbol dari *Use Case* Diagram adalah:

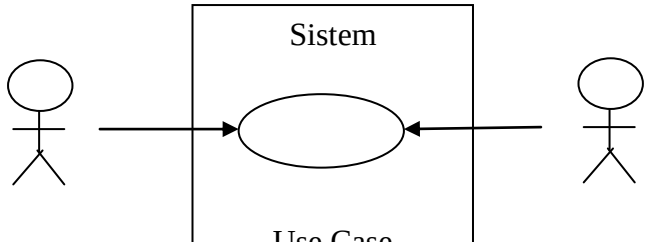
Tabel II.1. Simbol-simbol dari Use Case Diagram

Gambar	Keterangan

	Use case menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukan pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja awal nama use case
	Aktor adalah *abstraction* dari orang atau sistem yang lain yang mengaktifkan fungsi dari targert sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berintraksi dengan use case, tetapi tidak memiliki control terhadap use case
	Asosiasi antara aktor dan use case, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta intraksi secara langsung dan bukanya mengidikasikan aliran data.
	Asosiasi antara aktor dan use case yang menggunakan panah terbuka untuk mengidikasikan bila aktor berintaraksi secara pasif dengan sistem.
	Include, merupakan di dalam use case lain (required) atau pemanggilan use case oleh use case lain, contohnya adalah pemanggilan sebuah fungsi program.
	Extend, merupakan perluasan dari use case lain jika kondisi atau syarat terpenuhi.

(Sumber : Windu Gata ;2013 :4)

Berikut contoh dari *Use case* Diagram:



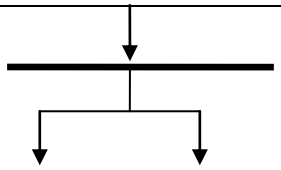
Gambar II.1. Contoh *Use case Diagram*

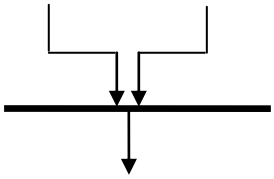
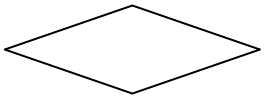
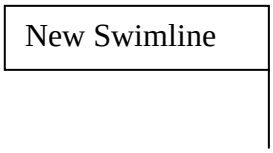
***(Sumber :* Adi Nugroho ;2010:86)**

II.3.1. *Activity Diagram*

Activity Diagram teknik untuk mendeskripsikan logika procedural, prose bisnis dan aliran kerja dalam banyak kasus. *Activity Diagram* mempunyai peran seperti halnya *flow chart*, akan tetapi perbedaanya dengan *activity diagram* adalah *activity diagram* bias mendukung perilaku parallel sedangkan *flowchart* tidak bisa. Berikut adalah sysmbol yang ada pada *activity diagram.*

Tabel II.2. Simbol Pada Activity Diagram

Gambar	Keterangan
	Start point, dilektakkan pada pojok kiri atas dan merupakan awal aktifitas
	End point, akhir aktifitas
	Activites, menggambarkan suatu proses/kegiatan bisnis.
	Fork(Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan parallel menjadi satu.

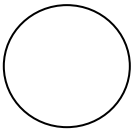
	Join(Penggabungan) atau rake, digunaknkan untuk menunjukkan adanya dekomposisi
	*Decision Points,* menggambarkan pilihan untuk pengambilan keputusan,true,false.
	*Swimlane,* pembagian *activity* diagram untuk menunjukkan siapa melakukan apa.

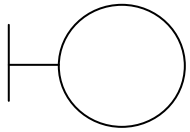
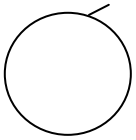
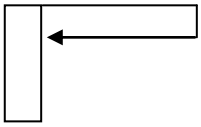
(Sumber : Windu Gata ;2013 :6)

II.3.2. *Sequence Diagram*

Sequence diagram menggambarkan intraksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. Sequence diagram biasa digunakan untuk menggambarkan scenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Simbol-simbol yang digunakan dalam *sequence diagram,* yaitu:

Tabel II.3. Simbol Pada *Sequence* Diagram

Gambar	Keterangan
	*Entity Class,*merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.

	*Boundary Class*,berisi kumpulan kelas yang menjadi *interface* antara satu atau lebih actor dengan sistem,seperti tampilan *formentry* dan *form* cetak.
	*Control class*, suatu objek yang berisi logika aplikasi tidak memiliki tanggung jawab kepada entitas,contohnya adalah kalkulasi dan aturan bisnis yang melibatkan barbagai objek.
	*Message*,simbol mengirim pesan antar *class.*
	*Recursive*,menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	*Activation*,*activation* mewakili sebuah eksekusi operasi dari panjang, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	Lifeline,garis-garis titik yang terhubung dengan objek,sepanjang *lifeline* terhadap *activation*

(Sumber : Windu Gata ;2013 :7)

II.3.3. *Class Diagram*

Class adalah hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem.

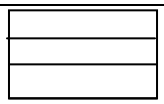
Class diagram juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan.

Class diagram secara khas meliputi: Kelas (*Class*), *Relasi*, *Associations*, *Generalization* dan *Aggregation*, Atribut(*Attributes*), Operasi (*Operations/Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut.

Adapun simbol-simbol yang terdapat pada Sequence diagram:

Tabel II.4. Simbol Pada *Sequence* Diagram

	Gambar	Keterangan

		Class,menambahkan kelas baru pada diagram
		Interface,menambahkan kelas antarmuka pada diagram
		Association,menggambarkan relasi asosiasi
		Generalization,mengambarkan relasi generalisasi
		Realize, menggambarkan relasi
		Aggregation,menggambarkan relasi agregasi

(Sumber : Windu Gata ;2013 :9)

Hubungan antar kelas mempunyai keterangan yang disebut dengan multiplicity atau kardinaliti

Tabel II.5. Kadinality Pada *Sequence* Diagram

Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada,maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 4

(Sumber : Windu Gata ;2013 :9)

II.3.4. Definisi UML

1. ***Unified Modeling Language***

Merupakan metode pengembangan perangkat lunak (sistem informasi) dengan menggunakan metode grafis serta merupakan bahasa untuk visualisasi, spesifikasi, konstruksi serta dokumentasi.

2. ***Unified Modeling Language (UML)***

   Adalah bahasa yang telah menjadi standard untuk visualisasi, menetapkan, membangun dan mendokumentasikan arti suatu sistem perangkat lunak.

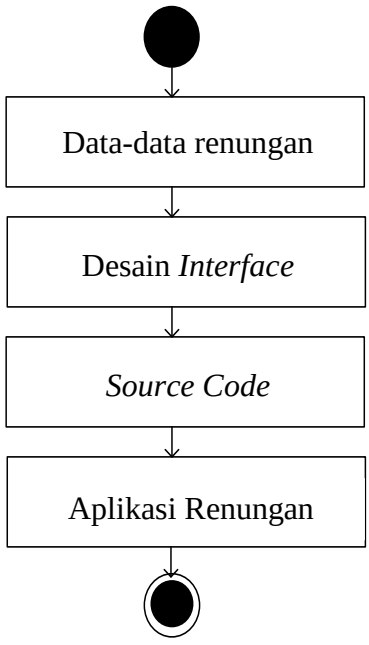
3. ***Unified Modeling Language (UML)***

   Dapat didefinisikan sebagai sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem perangkat lunak.

4. ***Unified Modeling Language (UML)***

   Merupakan standard modeling language yang terdiri dari kumpulan-kumpulan diagram, dikembangkan untuk membantu para pengembang sistem dan *software* agar bisa menyelesaikan tugas-tugas yang ada.

Berikut adalah *Unified Modeling Language (UML)* yang digunakan dalam merancang renungan harian kristiani berbasis multimedia.



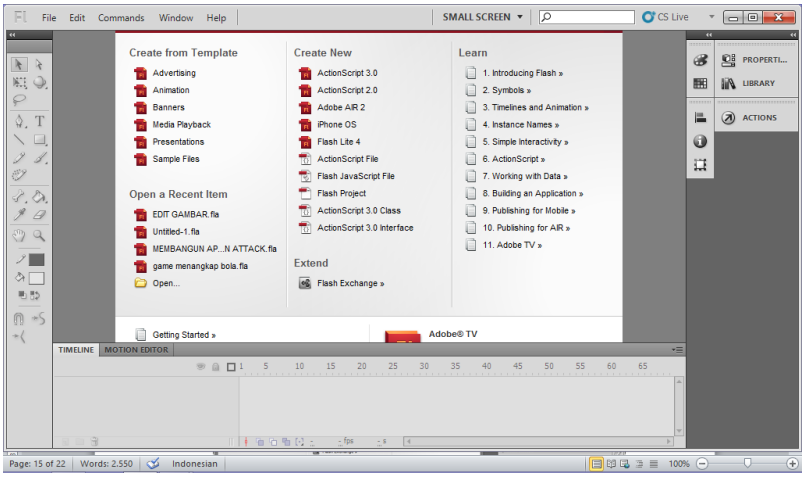
Gambar II.2. (UML) Aplikasi Renungan Harian Kristiani

II.4. Pengenalan Macromedia Flash CS5

Sebuah program grafis animasi standard propesional untuk menghasilkan produk-produk multi media seperti *Multimedia Persentation, wabsite, Computer Game,* dan *Animation.* Perogram ini mampu menghasilkan animasi yang begitu cangggih, sehingga besar aplikasi tutorial yang interaktif, game, presentasi, dan lain-lain dibuat dengan program ini. Flase CS5.5 merupakan pengembangan dari penyempurnaan dari virsi sebelumnya (*Flash Professional*). Ada beberapa panel pada plash yang harus diketahui sebagai dasar pembuatan animasi:

II.4.1. Area kerja macromedia Flash Player

Langkah untuk menjalankan program Adobe Flesh Pro CS5.5, tekan tombol **Start ► All Programs ► Adobe ► Adobe Flash CS5.5** sehingga tampil **Welcome Screen** seperti tanpak pada gambar berikut, (Madcoms Madium; 2012: 4).



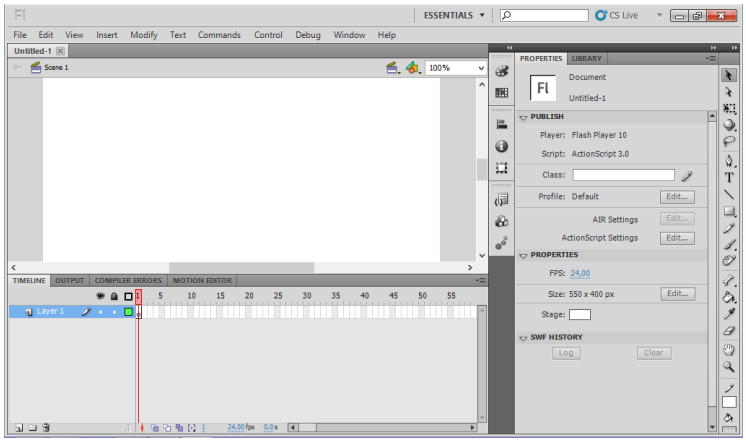
Gambar II.3. Tampilan layar pertama program Adobe Flash Pro CS.5.5

Sumber : (Madcoms Madium; 2012: 4).

Welcome Screen menampilka empat pilihan perintah untuk memulai Adobe Flash Pro CS5.5, yaitu:

1. ***Create from Template,*** berguna untuk membuka lembar kerja dengan template yang tersedia dalam program Adobe Flash Pro CS5.5.
2. ***Open a Recent Item,*** berguna untuk membuka kembali file yang pernah Anda simpan atau pernah Anda buka sebelumnya.
3. ***Create New,*** berguna untuk membuka lembar kerja baru dengan beberapa pilihan script yang tersedia.
4. ***Learn,*** berguna untuk membuka jendela Help yang berguna untuk mempelajari suatu perintah, (Madcoms Madium; 2012: 4-5).

Jika Anda tidak ingin menampilkan jendela *Welcome Screen* lagi saat membuka program, aktifkan kotak periksa ***Don't Show again*** yang terdapat pada sisi bawah dari jendela *Welcome Screen.* Sebagai cantoh klik perintah ***ActionScript 2.0*** pada bagian *Create New* sehingga tampil lembar kerja seperti Gambar II.4.



Gambar II.4. Tampilan Jendela program Adobe Flash Pro CS5.5

Sumber : (Madcoms Madium; 2012: 5).

II.4.2. Toolbox

Toolbox adalah sebuah panel yang menampung tombol-tombol yang berguna untuk membuat suatu desain animasi mulai dari tombol seleksi, pen, pensil, Text, 3D Rotation, dan lain-lain

II.4.3. Tampilan Panel

Panels, berisi kontrol fungsi yang di pakai dalam flash, yang berfungsi untuk mengganti dan memodifikasi berbagai atribut dari objek atau animasi secara cepat dan mudah. Panels biasanya terletak di bagian kanan area Flash. Untuk menampilkan panel tersebut, klik menu *Window>*(nama panel).

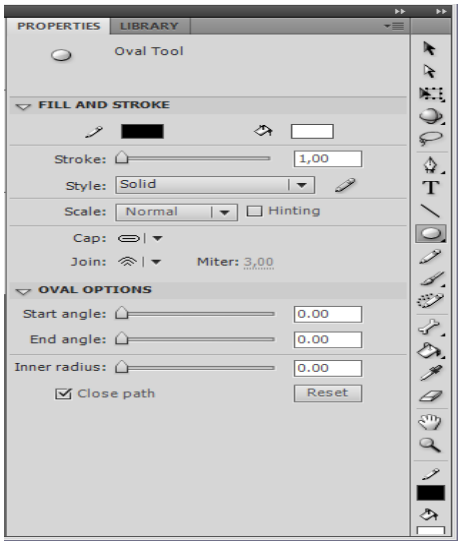
Program Adobe Flash Pro CS5.5 menampilkan lembar kerja yang sangaat menarik serta dapat di ubah menurut selera Anda. Dengan perubahan tersebut, Anda lebih mudah mengatur dan menyusun objek yang akan dianimasikan. Perhatikan gambar berikut serta keterangannya untuk mengatur tampilan lembar kerja Adobe Flash CS5.5,(Madcoms Madium; 2012: 14).

II.4.4. Timeline

Timeline berguna untuk menentukan durasi animasi, jumlah *layer, frame,* menempatkan *script* dan beberapa keperluan animasi lainnya. Semua bentuk animasi yang Anda buat akan diatur dan ditempatkan pada l*ayer* dalam *timeline.*

II.4.5. Panel Properties

Panel Properties berguna untuk menampilkan parameter dari sebuah tombol yang terpilih sehingga Anda dapat memodifikasi dan memaksimalkan fungsi dari tomol tersebut. Panel Propertis menampilkan parameter sesuai dengan tombol yang terpilih.

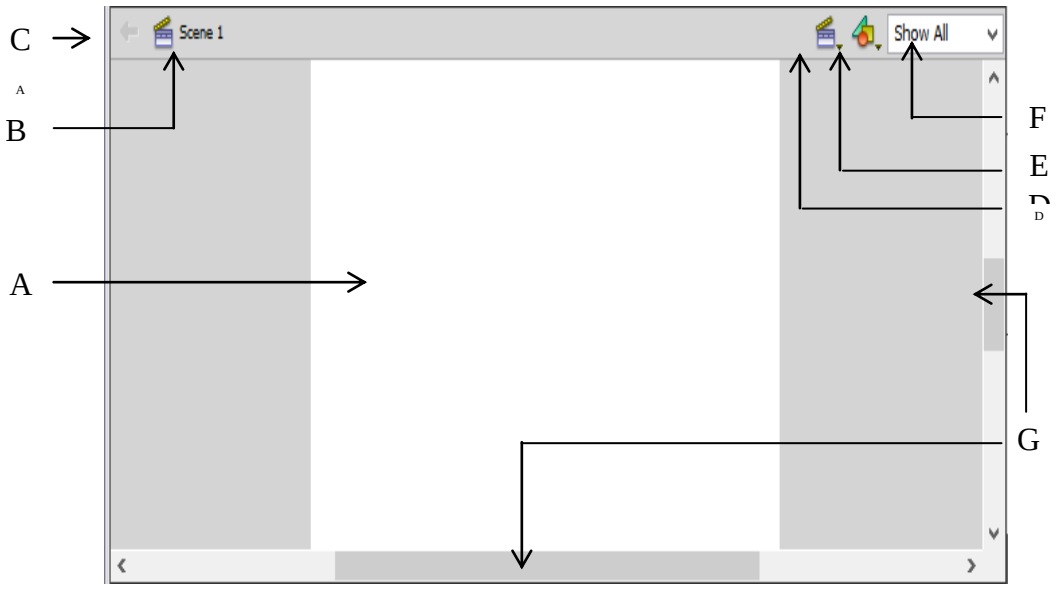


Gambar II.5. Tampilan Praperties

Sumber : (Madcoms Madium; 2012: 12).

II.4.6. Stage

Stage adalah lembar kerja yang di gunakan untuk membuat atau mendesain objek yang akan dianimasikan. Objek yang dibuat dalam lembar kerja dapat berupa objek *Vektor*, *Movie clip*, *Text*, *Button*, dan lain-lain, (Madcoms Madium; 2012: 11).



Gambar II.6. Tampilan Stage

Sumber : (Madcoms Madium; 2012: 11).

Tabel II.6. Keterangan Tampilan Stage

Abjad	Keterangan
A	**Stage**, lembar kerja untuk menyusun objek yang akan dianimasikan.
B	**Scene**, menunjukkan nama scene yang aktif.
C	Panah yang digunakan untuk berpindah dari lembar kerja simbol ke lember kerja utama.
D	**Edit Scene**, untuk memilih nama scane yang akan diedit.
E	**Edit Symbols**, untuk memilih nama simbol yang akan diedit.
F	**Zoom**, untuk mengatur besarnya tampilan stage atau lembar kerja.
G	**Scrolber**, untuk menggulung lembar kerja secara horisontal dan vertikal.

Sumber : (Madcoms Madium; 2012: 11).