BAB II
LANDASAN TEORI

II.1 Profil Instansi

Profil Badan Penanggulangan Bencana Daerah (BPBD) Provinsi Jawa Barat meliputi sejarah berdirinya BPBD, visi dan misi, logo BPBD, struktur organisasi serta tugas dan fungsi dari BPBD Provinsi Jawa Barat.

II.1.1 Sejarah Instansi

Begitu besarnya dampak yang ditimbulkan oleh gempa bumi 2 September 2009 membuat Pemerintah Provinsi Jawa Barat memahami bahwa penanganan bencana terutama yang berskala besar, tidak mungkin lagi ditangani dengan pola dan kelembagaan Satuan Koordinasi dan Pelaksanaan Penanggulangan Bencana (Satkorlak PB). Kesiapsiagaan seluruh lapisan di pemerintahan menjadi penting terkait dengan penanggulangan bencana itu. Apalagi, penanggulangan bencana merupakan suatu rangkaian kegiatan yang berkaitan dan memang kerap kali dilakukan secara lintas lembaga dan lintas wilayah. Koordinasi antar instansi terkait pun menjadi penting. Dari beberapa kejadian dan fakta yang ada bahwa bencana yang terjadi di Jawa Barat sangat beragam jenis dan skalanya. Jenis bencana yang berpotensi di Jawa Barat dapat dikelompokkan secara umum ada tiga jenis, yaitu bencana alam, bencana non alam, dan bencana sosial. Setiap bencana mengakibatkan dan berdampak terhadap kehilangan jiwa manusia, harta benda, sarana prasarana, serta psikologis korban bencana. Sebagaimana dijelaskan pada Pasal 5 Undang-Undang Nomor 24 Tahun 2007 tentang Penanggulangan Bencana, posisi pemerintah dan pemerintah daerah menjadi penanggung jawab dalam penyelenggaraan penanggulangan bencana. Karena menyadari kebutuhan-kebutuhan itulah, Badan Penanggulangan Bencana Daerah (BPBD) Provinsi Jawa Barat pun dibentuk pada tahun 2009. Tidak dapat dipungkiri, kejadian gempa bumi yang terjadi 2 September 2009 itu mempercepat langkah pembentukan BPBD Provinsi Jawa Barat. Sejak saat itu, keberadaan BPBD Jawa

Barat memiliki arti strategis terkait dengan penanggulangan bencana. Lembaga baru di lingkungan Pemerintah Provinsi Jawa Barat ini diharapkan bisa selalu siap siaga untuk menyelamatkan lebih banyak orang dan menekan kerugian harta dan benda para korban saat terjadi bencana.

Gambar II.1 Logo BPBD Provinsi Jawa Barat

Tugas dan fungsi BPBD Jawa Barat memang banyak. Tugas-tugas itu mulai dari sosialisasi pencegahan bencana, informasi tanggap bencana, kesiapsiagaan penyelamatan, sampai pada rehabilitasi serta rekonstruksi kawasan pascabencana. Dalam setiap tahapan kerja itu, tindakan BPBD Jawa Barat haruslah cepat, cermat, dan tepat karena berhubungan dengan nyawa dan masa depan manusia. Pada tahap awal, BPBD Jawa Barat sudah mulai berperan untuk menyebarkan informasi tanggap bencana, mensosialisasikan upaya pencegahan bencana, serta menginventarisasi daerah rawan bencana. Sejumlah tindakan pencegahan itu menjadi penting supaya masyarakat sudah terbentuk menjadi masyarakat siaga bencana. Upaya itu diharapkan dapat menekan kerugian harta benda dan terutama mencegah kehilangan nyawa yang merupakan dampak buruk akibat bencana. Saat bencana terjadi, BPBD Jawa Barat akan berperan melakukan penyelamatan, evakuasi korban, evakuasi harta benda korban, pemenuhan kebutuhan dasar para korban, dan pengurusan korban pengungsi. Pascabencana, BPBD Jawa Barat pun tetap berperan untuk melakukan rehabilitasi dan rekonstruksi rumah masyarakat yang menjadi korban.

Sedangkan tujuan dari BPBD Jawa Barat adalah untuk menjamin terselenggaranya penanggulangan bencana secara terencana, terpadu dan terkoordinasi dalam memberikan perlindungan kepada masyarakat dari ancaman, resiko & dampak bencana.

II.1.2 Visi dan Misi Instasi

Berikut ini merupakan visi dan misi dari BPBD Provinsi Jawa Barat :

a. Visi

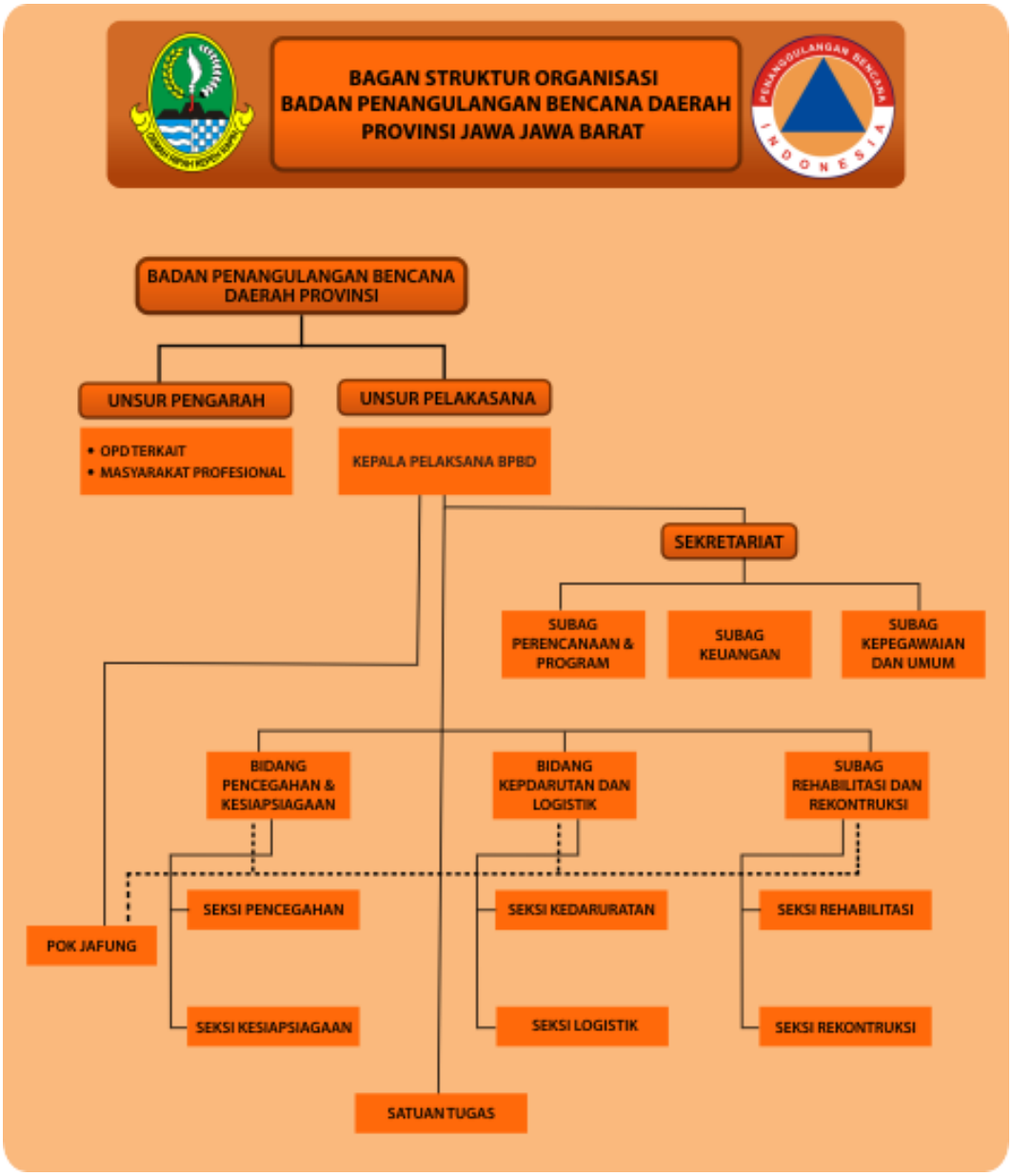
"Terciptanya penanggulangan bencana yang cepat, tepat dan profesional"

b. Misi

1. Terwujudnya penanggulangan bencana yang terencana, terpadu, dan terkodinasi.
2. Teciptanya jejaring komunikasi dan informasi.
3. Tesedianya masyarakat tanggap bencana berbasis pengurangan resiko bencana.
4. Tersedianya data berbasis rawan bencana.
5. Terwujudnya fasilitasi rehab rekon rumah masyarakat pasca bencana sesuai karakteristik dan kondisi alam setempat.

II.1.3 Struktur Organisasi

Struktur Organisasi BPBD Provinsi Jawa Barat :



Gambar II.2 Struktur Organisasi BPBD Provinsi Jawa Barat

II.2 Landasan Teori

Landasan Teori bertujuan memberikan gambaran dari teori yang terkait dalam pembangunan aplikasi. Landasan Teori yang dibahas yaitu Pengertian Sosialisasi, Media Sosialisasi, Multimedia, Metode yang digunakan, Bahasa Pemrograman yang digunakan dan *tools* yang digunakan.

II.2.1 Sosialisasi

Sosialisasi merupakan proses sosial tempat seorang individu mendapatkan pembentukan sikap untuk berprilaku yang sesuai dengan prilaku orang-orang disekitarnnya, sosialisasi juga dapat diartikan sebagai proses mempelajari kebiasaan dan tata kelakuan untuk menjadi satu bagian dari suatu masyarakat, sebagian adalah proses mempelajari peran [4].

Sosialisasi dapat disimpulkan sebagai proses belajar untuk penyesuaian diri di masyarakat yang berlangsung dalam proses yang lama dan bertahap seiring bertambahnya usia, tahapan proses sosialisasi dibagi enjadi 4 tahap yaitu [4]:

a. Tahap Pertama (*Preparatory Stage*)

   Tahap awal yang merupakan tahap persiapan individu untuk pertama kali mengenal lingkungan sosialnya, dimulai dari keluarga dan orang-orang tersekat seperti belajar berbicara dan berbahasa yang dicontohkan oleh ibu kepada anaknya.

b. Tahap Kedua (*Play Stage*)

   Tahap ini merupakan tahap lanjutan dari tahap pertama, pada tahap ini individu sudah dapat berkembang dengan lebih baik dan dapat memahami peran dirinya serta apa yang diharapkan dari dirinya dan peranan dari orang lain, seperti anak yang telah dapat membedakan status individu yang berada disekitarnya, seperti paman atau bibi dan kakek dan nenek.

c. Tahap Siap Bertindak (*Game Stage*)

Pada tahap ini individu telah memiliki ego berdasarkan kesadaran diri sendiri dan dapat bersikap mansiri, kemampuan untuk menyesuaikan diri semakin jelas dan kemampuan untuk menerima atau menyesuaikan dengan nilai norma yang berada di luar keluarganya pun dapat dijalaninya dengan kesadaran sebagai bagian aktif dari masyarakat.

d. Tahap Penerimaan Norma Kolektif (*Generelized Other*)

Tahap terakhir dimana individu telah dapat menempatkan diri pada masyarakat yang lebih luas lagi, memiliki kesadaran sebagai individu yang memiliki hak dan kewajiban, individu yang dapat dkenakan sanksi hukum apabila melanggar norma dan tata nilai di masyarakat.

II.2.2 Media Sosialisasi

Media sosialisasi (*Agent Of Socialization*) yang menjadi sarana bersosialisasi bagi setiap individu dapat dikategorikan dalam 4 media yaitu [4]:

1. Orang Tua dan Keluarga

Orang tua atau keluarga merupakan lingkungan pertama bagi seorang individu sejak lahir dan merupakan lingkungan yang paling lama ditinggali. Dalam lingkungan keluarga dikenal ada dua macam sosialisasi, yaitu :

a. Sosiaisasi Represif (*Represive Socialization*)

Sosialisasi ini menekankan serta mengharapkan ketaatan individu terhadap orang tuanya, orang tua menjadi pusat dan banyak melakukan dominasi terhadap proses belajar sosial individu tersebut.

b. Sosialisasi Partisipasi (*Participatiory Socialization*)

Sosialisasi ini berlawanan dengan Sosialisasi Represif karena menekankan pada individu sendiri sebagai pusat proses sosialisasi seperti adanya perhatian orang tua terhadap keinginan anaknya dan adanya imbalan pada anak apabila anak bertingkah dengan baik.

2. Teman Sepermainan

Teman sepermainan merupakan kelompok sosial yang jumahnya kecil memiliki kesamaan usia dan memiliki tujuan yang sama yaitu bermain atau memiliki tujuan rekreatif, dapat memberikan dampak positif dan negatif dalam perkembangan individu dalam kegiatan bersosialisasinya.

3. Sekolah

Pada lingkungan sekolah guru adalah pemegang kendali terhadap proses sosialisasi yang akan berlangsung dan bertujuan untuk meningkatkan pemahaman akan diri sendiri dan lingkungan seiring dengan proses pembelajaran di sekolah. Peranan sekolah tidak kalah pentingnya dengan orang tua dan keluarga sebab sekolah membantu individu yang bersekolah untuk menemukan jati dirinya serta memberikan bekal untuk berkembang dalam kehidupannya.

4. Media Massa

Merupakan salah satu unsur teknologi yang memiliki peran sebagai media sosialisasi, melelui media ini akan terjadi transformasi sosial serta budaya. Media ini dapat dijadikan sebagai alat untuk menyampaikan pesan tanpa terkait dengan norma yang ada di masyarakat, contohnya radio, koran, TV dan internet atau semua media yang membutuhkan daya listrik untuk mengoperasikannya.

II.2.3 Metode Pembelajaran Interaktif

Metode pembelajaran interaktif adalah sebuah metode pembelajaran yang digunakan untuk menyajikan suatu bahasan atau sumber pembelajaran yang dilakukan oleh guru untuk menciptakan suasana interaktif yang edukatif, antara guru dengan siswa, siswa dengan siswa dan dengan sumber pembelajaran yang mendukung terjadinya proses belajar dan mengajar. Metode Pembelajaran interaktif termasuk kedalam pembelajaran berbantuan komputer atau yang lebih dikenal dengan CAI (*Computer Assisted Instruction*) yaitu penggunaan komputer sebagai media penyampaian bahan ajar dengan melibatkan interaksi dari peserta didik serta memungkinkan adanya umpan balik atau suatu sistem penyampaian materi pelajaran berbasis komputer yang pelajarannya dirancang dan diprogram kedalam sistem tersebut [5].

CAI (*Computer Assisted Instruction*) dapat disajikan dengan cara :

a. Tutorial, penyajian materi secara bertahap dan untuk dipelajari secara berulang-ulang.

b. *Drills* and *Practice*, mencoba untuk meningkatkan kemahiran peserta didik dengan membantu siswa menguasai materi yang telah dipelajari sebelumnya dengan dilakukannya evaluasi tentang materi yang telah disampaikan sebelumnya.

c. Permainan dan Simulasi, menyajikan materi dengan cara yang lebih menarik dan memberikan kesempatan kepada peserta didik untuk bertindak sesuai dengan situasi tersebut.

d. Basis Data, menjadi sumber bagi siswa untuk membantu menambah informasi dan pengetahuan sesuai dengan keinginannya sendiri.

II.2.4 Multimedia

Multimedia berasal dari kata multi dan media, multi atau dalam bahasa latin disebut *nouns* yang berarti banyak dan media atau dalam bahasa latin disemut *medium* yang berarti perantara atau penghantar. Pengertian lainnya tentang multimedia, objek yang menjadi bagian multimedia yang terbagi dalam beberapa jenis yaitu [6] :

1. Teks

   Teks merupakan dasar dari pengolahan kata dan informasi berbasis multimedia yang bertujuan untuk menyampaikan/menyajikan informasi kepada pengguna. Selain itu teks juga merupakan sarana yang efektif untuk mengemukakan pendapat yang berupa ide-ide serta instruksi kepada pengguna.

2. Gambar

   Secara umum gambar berarti still image seperti foto, gambar merupakan salah satu sarana yang sangat baik untuk menyajukan informasi karena pengguna (manusia) sangat berorientasi pada visual / penglihatan.

3. Animasi

   Animasi adalah rangkaian gambar yang disusun secara berurutan atau dikenal dengan sebutan *frame*. Dalam satu *frame* terdiri dari beberapa atau banyak gambar, jiak susunan gambar tersebut ditampilkan secara tersusun dan dengan waktu tertentu maka susunan gambar tersebuta akan terlihat bergerak.

4. Audio

   Kelengkapan untuk lebih memperjelas penyampaian informasi. Suara juga dapat menjelaskan karakteristik suatu gambar seperti *sound effect*.

5. Video

   Video adalah elemen multimedia yang paling kompleks karena penyampaian informasi yang lebih komunikatif dibandingkan dengan gambar biasa. Video berbeda dengan animasi dilihat dari penyajiannya yang utuh dalam kesatuan dari objek yang dimodifikasi sehingga terlihat saling mendukung penggambaran yang seakan-akan terlihat hidup.

6. *Interactive Link*

   Menggabungkan beberapa elemen multimedia unuk menciptakan informasi yang terpadu untuk pengaksesan informasi pada multimedia (linear dan non-linear). Sebagian multimedia adalah interakti, dimana pengguna dapat menekan mouse atau objek pada layar seperti tombol atau teks yang menyebabkan sistem menjalankan perintah tertentu.

II.2.5 Aplikasi Berbasis Dekstop

Aplikasi berbasis *desktop* adalah aplikasi yang dapat berjalan sendiri (independen) tanpa menggunakan *browser* ataupun internet. Berdasarkan informasi yang didapat dari *Informations Communications Group and Microsoft Corporation* [7]. Aplikasi Berbasis Desktop memiliki beberapa kekurangan dan kelebihan sebagai berikut :

1. Kekurangan Aplikasi Berbasis Desktop adalah :
   a. Sulit dalam melakukan akses jarak jauh.
   b. Sulit dalam melakuakan distribusi dalam jumlah yang sangat besar karena harus melakukan proses installasi pada setiap titik distribusi.
   c. Proses *update* terbilang sulit.
2. Kelebihan Aplikasi Berbasis Desktop adalah :
   a. Kinerja umumnya lebih baik dibandingkan dengan Aplikasi Berbasis Web kerena disesuaikan dengan spesifikasi perangkat keras yang ada.

b. Peroses pembangunannya lebih mudah disbanding dengan proses pembangunan Aplikasi Berbasis Web yang menggunakan HTML.

II.2.6 *Object Oriented Programing* (OOP)

Object Oriented Programming (OOP) atau yang dikenal dengan Pemrograman Berorientasi Objek merupakan cara membangun aplikasi/sistem yang berorientasikan kepada objek. Objek sendiri adalah sesuatu yang memiliki satu atau lebih atribut (*Field*) dan macam aksi yang terdapat didalamnya (*Method*).

Model data berorientasi objek dikatakan dapat memberi fleksibilitas yang lebih, kemudahan mengubah program, dan digunakan luas dalam teknik piranti lunak skala besar. Lebih jauh lagi, pendukung OOP mengklaim bahwa OOP lebih mudah dipelajari bagi pemula dibanding dengan pendekatan sebelumnya, dan pendekatan OOP lebih mudah dikembangkan dan dirawat [8].

Dengan menggunakan OOP maka dalam melakukan pemecahan suatu masalah kita tidak melihat bagaimana cara menyelesaikan suatu masalah tersebut (terstruktur) tetapi objek-objek apa yang dapat melakukan pemecahan masalah tersebut. Untuk menyelesaikan suatu masalah dengan kolaborasi antar objek-objek yang ada karena setiap objek memiliki deskripsi tugasnya sendiri. Pemrograman orientasi-objek menekankan konsep berikut [8] :

a. *Class*

   Kumpulan atas definisi data dan fungsi-fungsi dalam suatu unit untuk suatu tujuan tertentu. Sebuah class adalah dasar dari modularitas dan struktur dalam pemrograman berorientasi objek (*Class* juga dapat berarti sketsa/*blueprint* dari sebuah objek). Secara umum *Class* memiliki 2 anggota yaitu *Field* atribut yang dimiliki oleh objek dari sebuah *Class* dan *Method* aksi yang dapat dilakukan oleh objek dari sebuah *Class*. Sebagai contoh terdapat *Class of Horse* yang memiliki

atribut berkaki 4, bermata 2 dan memiliki 1 ekor, selain itu aksi yang dimiliki oleh *Class* tersebut adalah tidur, makan dan berjalan.

b. Objek

Adalah realisisi dari sebuah *Class* yang didalamnya terdapat atribut (*Field*) dan aksi (*Method*) yang menjadi suatu unit (*Class*) dalam sebuah program komputer. Objek merupakan dasar dari modularitas dan struktur dalam sebuah program komputer berorientasi objek.

c. Abstraksi

Sebuah cara untuk membuat suatu *Class* yang memiliki keturunan (Inheritas) agar tidak dapat tercipta objek dari *Class* tersebut dengan tujuan menjadikan *Class* tersebut menjadi pola bagi *Class* turunannya. Sebagai contoh terdapat sebuah *Class* yang memiliki dua keturunan yang nantinya akan dibuat objek yang tercipta dari keturunannya yang tentunya dapat memiliki atribut dan aksi yang dimiliki oleh *Class* induk. Jika tidak dilakukan abstraksi pada *Class* induk maka dapat dibuat sebuah objek yang yang dapat memiliki atribut dan aksi yang tidak dimiliki oleh keturunan dari *Class* induk, seharusnya *Class* induk hanyalah sebuah bentuk umum/pola yang nantinya dapat digunakan oleh keturunannya tidak untuk dijadikan sebuah objek.

d. Enkapsulasi

Sebuah cara yang dilakukan untuk menjaga keamanan dan fleksibilitas dari sebuah *Class* yang dapat dilakukan dengan (membungkusnya) menggunakan *Access Identifier* yang dapat menentukan siapa saja yang dapat mengakses (baik dibaca maupun ditulis/diubah) anggota dari sebuah *Class*, *Method Getter/Setter* yang menjadikan sifat dari anggota tertentu pada sebuah *Class* menjadi dapat dibaca oleh *Class* lain namun hanya bisa ditulis/diubah oleh dirinya sendiridan *Property* yang merupakan cara yang lebih praktis dibandingkan dengan *Method Getter/Setter* dikarenakan *Property* membuat method *getter* dan *setter*

menjadi satu tidak seperti *Method Getter/Setter* yang memisahkan antara *Method Getter* dan *Setter*.

e. Polimorfisme

Polimorfisme atau dalam bahasa Yunani disebut *Polymorphysm* yang berarti banyak bentuk. Polimorfisme sendiri memiliki dua karakteristik yaitu :

1. Sebuah pointer dari *Class* tertentu tidak hanya dapat mengacu pada objek yang tercipta dari dirinya sendiri pointer tersebut juga dapat menampung objek dari *Class* keturunannya (sebuah pointer dari suatu Class dapat mengacu kepada benyak jenis objek).
2. Sebuah method yang terdapat pada sebuah *Class* dapat memiliki banyak implementasi, sebagai contoh terdapat sebuah *Class* yang menulis ulang (*override*) *method* yang diturunkan kepadanya.

f. Inheritas

Mengatur polimorfisme dan enkapsulasi dengan mengijinkan objek didefinisikan dan diciptakan dari *Class* induk dengan jenis khusus/berbeda dari objek yang sudah ada objek-objek ini dapat membagi (dan memperluas) perilaku mereka tanpa harus mengimplementasi ulang perilaku tersebut (bahasa berbasis-objek tidak selalu memiliki inheritas).

II.2.7 Unified Modelling Language (UML)

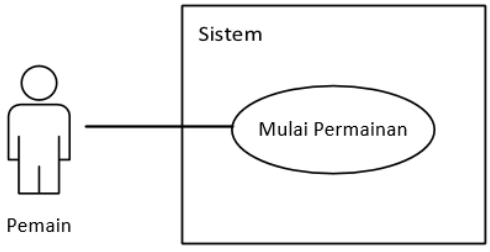
Unified Modelling language (UML) adalah keluarga notasi grafis yang didukung oleh meta-model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek (OO). Definisi ini merupakan definisi yang sederhana. Pada kenyataanya, pendapat orang-orang tentang UML berbeda satu sama lain. Hal ini dikarenakan oleh sejarahnya sendiri dan oleh perbedaan

persepsi tentang apa yang membuat sebuah proses rancang-bangun perangkat lunak efektif [9].

Bahasa pemrograman grafis telah ada di industri perangkat lunak sejak lama. Pemicu utaman di balik semuanya adalah bahasa pemrograman berada pada tingkat abstraksi yang tidak terlalu tinggi untuk diskusi tentang desain. UML merupakan standar yang relatif terbuka yang dikontrol oleh *Object Management Group* (OMG), sebuah konsorsium terbuka yang terdiri dari banyak perusahaan. OMG dibentuk untuk membuat standar-standar yang mendukung interoperabilitas, khususnya interoperabilitas sistem berorientasi objek. OMG mungkin lebih dikenal dengan standar-standar CORBA (*Commom Object Request Broker Architecture*). Uml lahir dari penggabungan banyak bahasa pemodelan grafis berorientasi objek yang berkembang pesat pada akhir 1980-an dan awal 1990-an.

II.2.7.1 Use Case Diagram

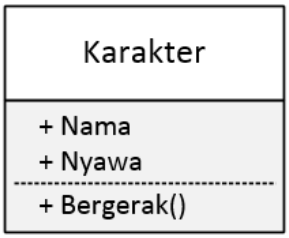
Use case adalah teknik untuk merekam persyaratan fungsional sebuah sistem. Sebuah *use case* juga menggambarkan cara sistem berperilaku untuk memenuhi kebutuhan. *Use case* mendeskripsikan interaksi tipikal antara para pengguna sistem dengan sistem itu sendiri, dengan memberi sebuah narasi tentang sistem tersebut digunakan. Dalam bahasan Use case, para pengguna disebut sebagai aktor. Aktor merupakan sebuah peran yang dimainkan seorang pengguna dalam kaitannya dengan sistem. Aktor tidak harus manusia, jika sebuah sistem melakukan sebuah layanan untuk sistem komputer lain, sistem lain tersebut merupakan aktor [10].



Gambar II.3 Contoh *Usecase Diagram* [10]

II.2.7.2 Class Diagram

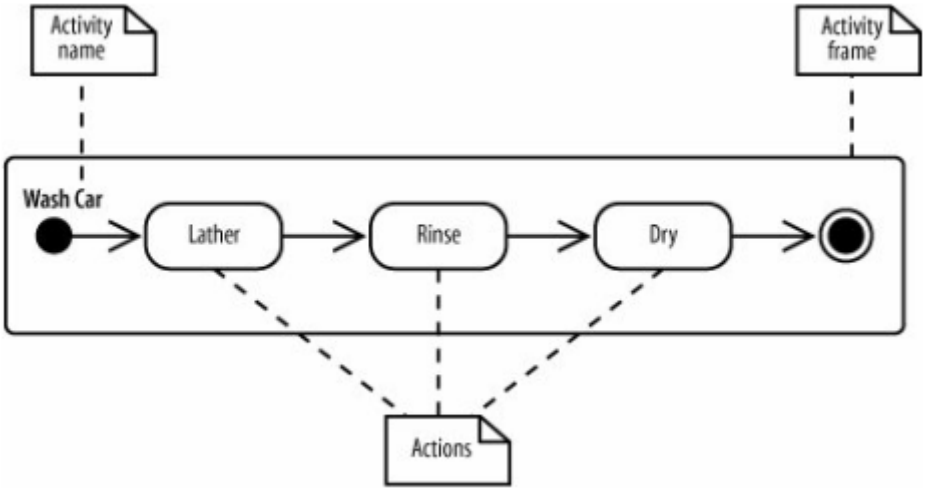
Class diagram mendeskripsikan jenis-jenis objek dalam sistem dan berbagai macam hubungan statis yang terdapat di antara mereka. *Class diagram* juga menunjukan properti dan operasi sebuah class dan batasan-batasan yang terdapat dalam hubungan-hubungan objek tersebut. UML menggunakan istilah fitur sebagai istilah umum yang meliputi properti dan operasi sebuah class [10].



Gambar II.4 Contoh *Class Diagram* [10]

II.2.7.3 Activity Diagram

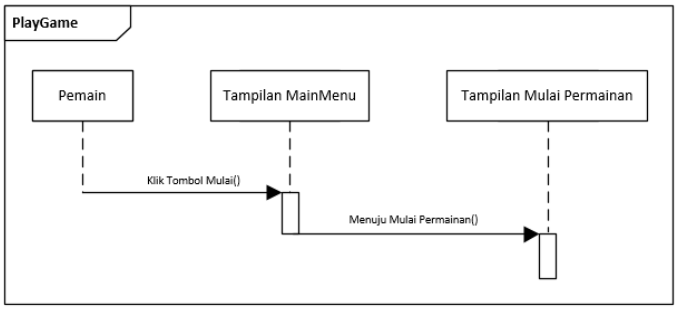
Activity diagram adalah teknik untuk menggambrakan logika prosedural, proses bisnis, dan jalur kerja. Dalam beberapa hal, diagram ini memainkan peran mirip sebuah diagram alir, tetapi perbedaan prinsip antar diagram ini dan notasi diagram alir adalah diagram ini mendukung sifat pararel. *Activity diagram* memungkinkan siapapun yang melakukan proses untuk memilih urutan dalam melakukannya. Dengan kata lain, diagram hanya menyebutkan aturan-aturan rangkaian dasar yang harus kita ikuti. Hal ini penting untuk pemodelan bisnis, karena proses-proses sering muncul secara pararel. Ini juga berguna pada algoritma yang bersamaan, di mana urutan-urutan independen dapat melakukan hal-hal secara pararel Contoh sederhana penggunaan *activity diagram* pada kasus proses mencuci mobil dapat dilihat pada gambar II.5 [10].



Gambar II.5 Contoh *Activity Diagram* [10]

II.2.7.4 Sequence Diagram

Sequence Diagram menunjukan interaksi antar diagram yang berkolaborasi dalam beberapa sifat. Sebuah *sequence diagram* secara khusus menjabarkan sifat skenario tunggal. Diagram tersebut menunjukan sejumlah objek contoh dan pesan-pesan yang melewati objek - objek ini di dalam *Use case* [10].



Gambar II.6 Contoh *Sequence Diagram* [10]

II.2.8 Action Script 3.0

Action Script adalah bahasa pemrograman yang dibuat berdasarkan ECMAScript, digunakan dalam pengembangan situs web dan perangkat lunak menggunakan platform Adobe Flash Player. Bahasa ini awalnya dikembangakan oleh Macromedia, akan tetapi sekarang sudah dimilki dan dilanjutkan perkembangannya oleh Adobe, yang membeli Macromedia pada tahun 2005.

Action Script 3.0 merupakan pengambangan dari versi sebelumnya yaitu action script 2.0 yang dirasa kurang cepat dalam performa dan lambat dalam pengolahan data. Actionscript 3.0 memiliki fitur terbaru yaitu [11] :

1. *Runtime Types*
   Dalam ActionScript 2.0, tipe anotasi terutama bantuan untuk pengembang saat *runtime*, semua nilai yang *dynamically typed*. Dalam ActionScript 3.0, tipe informasi yang disimpan saat *runtime* dan dimanfaatkan untuk beberapa tujuan. *Flash Player runtime* melakukan pemeriksaan pengetikan, meningkatkan sistem keamanan. Jenis informasi ini juga digunakan untuk mewakili variabel dalam mesin asli representasi, meningkatkan kinerja dan mengurangi penggunaan memori.

2. *Sealed Classes*
   ActionScript 3.0 memperkenalkan konsep kelas yang tertutup rapat (*Encapsulation*). Sebuah kelas disegel hanya memiliki seperangkat tetap properti dan metode yang didefinisikan pada saat kompilasi properti dan metode tambahan tidak dapat ditambahkan. Hal ini membuat waktu kompilasi yang lebih ketat, menghasilkan program yang lebih kuat. Ini juga meningkatkan penggunaan memori dengan tidak memerlukan tabel hash internal untuk setiap objek misalnya. Kelas dinamis juga mungkin menggunakan kata kunci *dynamic*.

3. *Method Closures*
   Event handling disederhanakan dalam ActionScript 3.0 berkat method closures, yang menyediakan *built-in* delegasi. Dalam ActionScript 2.0, closure tidak akan ingat objek itu berasal, menyebabkan perilaku tak terduga ketika closure itu digunakan. Kelas *mx.utils.Delegate* solusi yang populer, untuk menggunakannya, perlu menulis kode sebagai berikut :

   myButton.addEventListener("click", Delegate.create(this, someMethod));

   Delegate.create(this, someMethod);

Kelas ini tidak lagi diperlukan karena dalam ActionScript 3.0, sebuah metode penutupan akan dihasilkan ketika *someMethod* adalah ditunjuk. Metode *Closures* akan secara otomatis mengingat objek aslinya misalnya:

myButton.addEventListener("click", someMethod);

4. *ECMAScript* Untuk XML (E4X)
   ActionScript 3.0 mengimplementasi penuh ECMAScript untuk XML (E4X), baru-baru ini sebagai standar ECMA-357. E4X secara natural menawarkan fasih bahasa set konstruksi untuk memanipulasi XML. Tidak seperti *tradisional parsing* XML API, E4X membuat XML merasa seperti tipe data asli dari bahasa. E4X mengeffektifkan pengembangan aplikasi yang memanipulasi XML dengan secara drastis mengurangi jumlah kode yang dibutuhkan.

5. *Regular expressions*
   ActionScript 3.0 meliputi dukungan asli untuk ekspresi reguler sehingga Anda dapat dengan cepat mencari dan memanipulasi string. ActionScript 3.0 mengimplementasikan ekspresi reguler didefinisikan dalam Bahasa Spesifikasi ECMAScript (ECMA-262).

6. *Namespaces*
   Namespaces adalah mekanisme baru yang inovatif untuk mengendalikan visibilitas deklarasi. Serupa dengan *specifiers* akses tradisional yang digunakan untuk mengontrol visibilitas deklarasi (*public, private, protected*), ruang nama *custom* dasarnya *specifiers* akses, yang dapat memiliki nama yang dipilih. Flex framework, misalnya, menggunakan *namespace* untuk mx_internal data internal. *Namespaces* dilengkapi dengan *Universal Resource Identifier* (URI) untuk menghindari bentrokan, dan juga digunakan untuk mewakili *XML namespaces* ketika bekerja dengan E4X.

7. *New Primitive Types*
   ActionScript 2.0 memiliki tipe numerik tunggal, Number ganda presisi *floating point* nomor. Suatu penambahan ActionScript 3.0 yang baru int

tipe 32-bit integer yang memungkinkan penandaan kode ActionScript mengambil keuntungan dari kemampuan matematika integer cepat dari CPU. *Type data* int bagus untuk *loop counter* dan hampir di mana saja titik desimal tidak diperlukan. Jenis baru lainnya uint sebuah 32-bit *unsigned integer* mirip dengan tipe int.

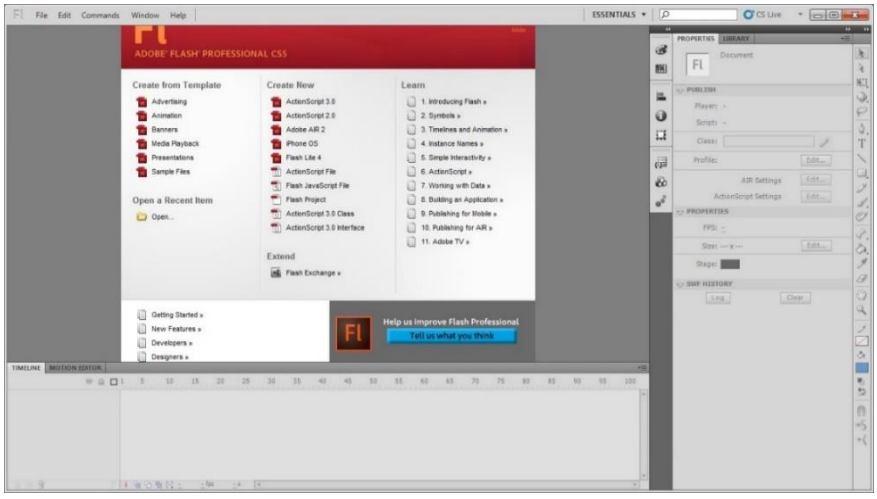
8. *Flash Player API features*
   Flash Player API adalah kumpulan kelas dan fungsi yang mengekspos kemampuan Flash Player dengan bahasa ActionScript. Fungsi ini adalah jembatan antara inti/*core* bahasa ActionScript dan sisanya dari *platform*. Ini adalah sumber dari banyak keunggulan yang tersedia untuk aplikasi Flash dan sangat penting untuk melengkapi bahasa inti. Meskipun tidak ada ruang di sini untuk menutupi API secara rinci.

9. *DOM3 Event Model*
   Model event menyediakan cara standar untuk menghasilkan dan penanganan pesan *event* sehingga objek di dalam aplikasi dapat berinteraksi dan berkomunikasi, menjaga status dan merespon perubahan. Berpola setelah W3C Acara DOM3 spesifikasi, model ini memberikan gambaran yang lebih jelas dan lebih efisien daripada acara mekanisme sistem yang tersedia dalam versi sebelumnya ActionScript. Framework aplikasi Flex menggunakan model peristiwa yang sama seperti Flash Player API, sehingga sistem acara dipersatukan di platform dari atas ke bawah.

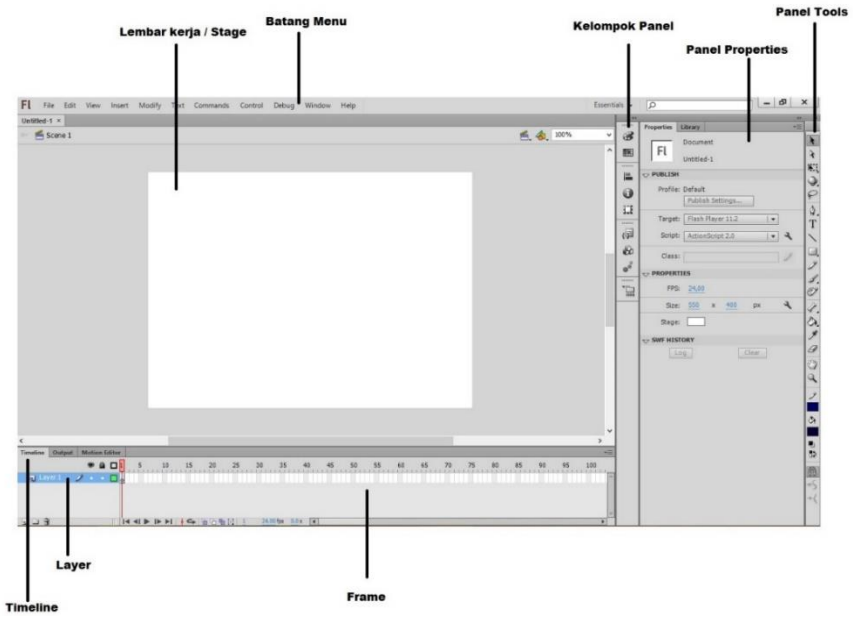
II.2.9 Adobe Flash CS5

Adobe Flash Professional CS5 merupakan salah satu program untuk membuat animasi 2D vektor. Dalam perkembangannya, program Adobe Flash Pro melakukan banyak penyempurnaan pada setiap versinya. Program Adobe Flash Pro versi CS5 ini, memiliki beberapa fitur baru yang membuat Adobe Flash semakin baik untuk mengolah animasi 2D berbasis vektor.

Dengan Program Adobe Flash Professional CS5, kita dapat membuat berbagai aplikasi animasi 2D mulai dari animasi kartun, animasi interaktif, game, company profile, presentasi, video clip, animasi web dan aplikasi animasi lainnya sesuai dengan kebutuhan. Sekarang Adobe Flash didukung dengan kemampuannya dalam membuat animasi transformasi 3D, serta panel yang khusus untuk pembuatan animasi dengan ActionScript yang lebih mudah pemakaiannya. Berikut merupakan tampilan dari Adobe Flash Professional CS5 yang dapat dilihat pada gambar II.7 dan tampilan dari lembar kerja Adobe Flash Professional CS5 yang dapat dilihat pada gambar II.8.



Gambar II.7 Halaman Awal Adobe Flash CS5



Gambar II.8 Lembar Kerja Adobe Flash CS5

Beberapa komponen kerja program Adobe Flash Pro CS5 telah ditampilkan sebagai tampilan standart pada Gambar II.8 Berikut ini adalah beberapa komponen kerja dari *Adobe flash Pro CS5* :

1. *Toolbox* adalah sebuah panel yang menampung tombol-tombol yang berguna untuk membantu suatu desain animasi mulai dari tombol seleksi, *pen*, pensil, text, 3D *rotation*, dan lain-lain.
2. *Timeline* berguna untuk menentukan durasi animasi, jumlah layer, frame, menempatkan *script* dan beberapa keperluan animasi lainnya. Semua bentuk animasi yang dibuat akan diatur dan ditempatkan pada layer dalam *timeline*.
3. *Stage* adalah lembar kerja yang digunakan untuk membuat atau mendesain objek yang akan dianimasikan.
4. *Panel properties* berguna untuk menampilkan parameter dari sebuah tombol yang terpilih sehingga dapat memodifikasi dan memaksimalkan fungsi dari tombol tersebut.
5. *Efek filter* adalah bagian dari panel propertis yang menampilkan berbagai jenis *efek filter* yang dpat digunakan untuk mempercantik tampilan objek.

6. *Motion editor* berguna untuk melakukan kontrol animasi yang telah dibuat. *Panel motion presets* menyimpan format animasi yang telah jadi dan siap digunakan sewaktu-waktu jika diperlukan.