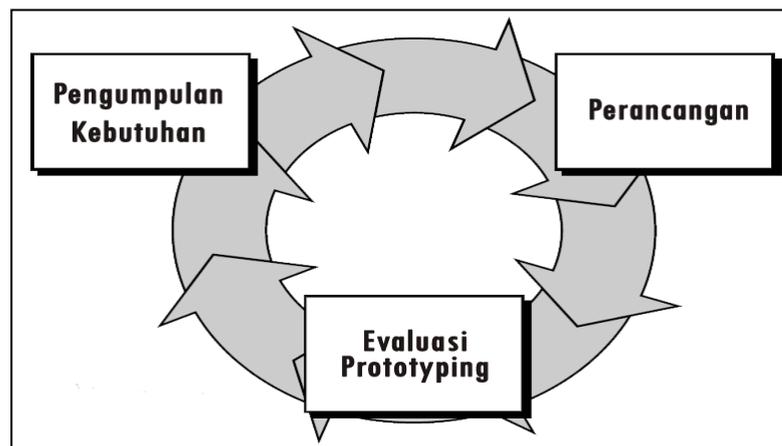


BAB II LANDASAN TEORI

2.1 Metode Pengembangan Perangkat Lunak

2.1.1 Model *Prototype*

Pressman (2012, p.40) menyatakan bahwa *Prototyping* paradigma dimulai dengan pengumpulan kebutuhan, pengembang bertemu dengan pengguna dan mengidentifikasi objektif keseluruhan dari perangkat lunak, selanjutnya mengidentifikasi segala kebutuhan yang diketahui secara garis besar di mana definisi-definisi lebih jauh merupakan keharusan, kemudian dilakukan perancangan kilat, lalu diakhiri dengan evaluasi prototyping yang dapat dilihat pada gambar 2.9 berikut :



Gambar 2.9 Model *Prototype*.

Tahap–tahap rekayasa *software* dalam *prototype model* pada gambar 2.9 di atas adalah sebagai berikut :

1) Pengumpulan kebutuhan

Developer dan klien bertemu untuk menentukan tujuan umum, kebutuhan yang diketahui dan gambaran bagian-bagian yang akan dibutuhkan berikutnya.

Detail kebutuhan mungkin tidak dibicarakan disini, pada awal pengumpulan kebutuhan. Selanjutnya peneliti akan melakukan analisis terhadap data apa

saja yang dibutuhkan, seperti analisis terhadap sistem yang berjalan, analisis kebutuhan perangkat lunak, analisis kebutuhan perangkat keras, dan analisis kebutuhan materi pembelajaran.

2) Perancangan

Perancangan dilakukan dengan cepat dan rancangan mewakili semua aspek software yang diketahui, dan rancangan ini menjadi dasar pembuatan *prototype*. Dalam tahap ini peneliti akan membangun sebuah versi *prototype* yang dirancang kembali dimana masalah-masalah tersebut diselesaikan

3) Evaluasi *prototype*

Pada tahap ini, calon pengguna mengevaluasi *prototype* yang dibuat dan digunakan untuk memperjelas kebutuhan *software*. *Software* yang sudah jadi dijalankan dan akan dilakukan perbaikan apabila kurang memuaskan. Perbaikan termasuk dalam memperbaiki kesalahan / kerusakan yang tidak ditemukan pada langkah sebelumnya.

Kelebihan dari *Prototype Model* adalah sebagai berikut :

- 1) End user dapat berpartisipasi aktif.
- 2) Penentuan kebutuhan lebih mudah diwujudkan.
- 3) Mempersingkat waktu pengembangan *software*.

Kekurangan dari *Prototype Model* adalah sebagai berikut:

- 1) Proses analisis dan perancangan terlalu singkat.
- 2) Mengesampingkan alternatif pemecahan masalah.
- 3) Biasanya kurang fleksibel dalam menghadapi perubahan.
- 4) *Prototype* yang dihasilkan tidak selamanya mudah dirubah.
- 5) *Prototype* terlalu cepat selesai.

2.2 UML (*Unified Modeling Language*)

2.2.1 Pengertian UML

Menurut Rossa dan Shalahuddin, UML (*Unified Modeling Language*) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek.

UML merupakan bahasa untuk membangun dan mendokumentasikan *artifacts* (bagian dari informasi yang digunakan atau dihasilkan oleh proses pembuatan perangkat lunak, *artifact* tersebut dapat berupa model, deskripsi atau perangkat lunak) dari sistem perangkat lunak, seperti pada pemodelan bisnis dan sistem non perangkat lunak. Selain itu UML adalah bahasa pemodelan yang menggunakan konsep orientasi object. UML dibuat oleh Grady Booch, James Rumbaugh, dan Ivar Jacobson di bawah bendera Rational Software Crop. UML menyediakan notasi-notasi yang membantu memodelkan sistem dari berbagai perspektif. UML tidak hanya digunakan dalam pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan.

2.2.2 Bagian-Bagian UML

Bagian-bagian utama dari UML adalah *view*, diagram, model element, dan *general mechanism*. Diagram berbentuk grafik yang menunjukan simbol elemen model yang disusun untuk mengilustrasikan bagian atau aspek tertentu dari sistem. Sebuah diagram merupakan bagian dari suatu *view* tertentu dan ketika digambarkan biasanya dialokasikan untuk *view* tertentu. Adapun jenis diagram antara lain:

1) *Use Case Diagram*

Use case adalah abstraksi dari interaksi antara system dan actor. *Use case* bekerja dengan cara mendeskripsikan tipe intraksi antara lain user sebuah system dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah system dipakai. *Use case* merupakan konstruksi untuk mendeskripsikan

bagaimana sistem akan terlihat di mata user. Sedangkan *Use case* diagram memfasilitasi komunikasi diantara analis dan pengguna serta analis dan client.

2) *Class Diagram*

Class adalah deskripsi kelompok obyek-obyek dengan *property*, perilaku (operasi) dan relasi yang sama. Sehingga dengan adanya *Class diagram* dapat memberikan pandangan global atas sebuah system. Hal tersebut tercermin dari *class-class* yang ada dan relasinya satu dengan yang lainnya. Sebuah sistem biasanya mempunyai beberapa *class diagram*. *Class diagram* sangat membantu dalam visualisasi setruktur kelas dari suatu sistem.

3) *Activity Diagram*

Menggambarakan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti *use case* atau intraksi.

2.2.3 Tujuan dan Keunggulan UML

Tujuan UML adalah Memodelkan suatu sistem (bukan hanya perangkat lunak) yang menggunakan konsep berorientasi object, menciptakan suatu bahasa pemodelan yang dapat digunakan baik oleh manusia maupun mesin.

Keunggulan menggunakan UML dibandingkan menggunakan metodologi terstruktur :

1) *Uniformity*

Pengembangan cukup menggunakan satu metodologi dari tahap analisis hingga perancangan. Memungkinkan merancang komponen antarmuka secara terintegrasi bersama perancangan perangkat lunak dan perancangan struktur data.

2) *Understandability*

Kode yang dihasilkan dapat diorganisasi kedalam kelas-kelas yang berhubungan dengan masalah yang sesungguhnya sehingga lebih mudah untuk dipahami.

3) *Stability*

Kode program yang dihasilkan relatif stabil sepanjang waktu, karena mendekati permasalahan yang sesungguhnya.

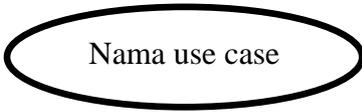
4) *Reusability*

Dengan metodologi berorientasi objek, dimungkinkan penggunaan ulang kode, sehingga pada akhirnya akan sangat mempercepat waktu pengembangan perangkat lunak (atau sistem informasi).

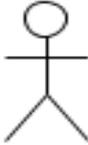
2.2.4 Simbol-Simbol pada UML

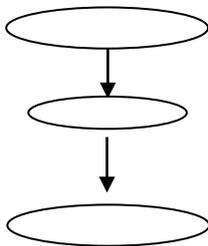
Simbol-simbol yang terdapat dalam diagram UML. Dapat dilihat pada tabel 2.1 dibawah ini :

Tabel 2.1 Simbol Pada *Diagram* UML.

Simbol	Deskripsi
<p><i>Use Case</i></p> 	<p>Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan anatar unit atau aktor, biasanya akan diterangkan dengan menggunakan kata kerja diawal-diawal frase nama <i>use case</i>.</p>

Tabel 2.1 Simbol Pada *Diagram* UML (Lanjutan).

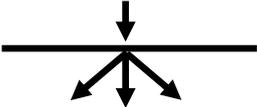
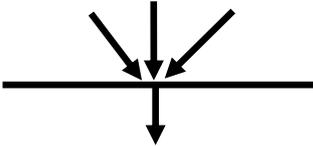
<p>Aktor/Actor</p> 	<p>Orang, proses, atau sistem lain yang berintraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol</p>
---	---

<p>Nama Aktor</p>	<p>dari aktor adalah orang, tapi aktor belum tentu merupakan orang. Biasanya akan dinyatakan menggunakan kata benda diawal <i>frase</i> nama aktor.</p>
<p>Asosiasi/Association</p> <p>_____</p>	<p>Komunikasi antar aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.</p>
<p>Ekstensi/Extend</p> <p><<extend>></p> 	<p>Case tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, misal .</p>  <p>Arah panah mengarah pada <i>use case</i> yang ditambahkan.</p>
<p>Uses</p> <p><<uses>></p>	<p>Digunakan sebagai kegiatan utama atau syarat menuju <i>use case</i> berikutnya.</p>

2.2.5 Activity Diagram

Diagram aktivitas digunakan untuk menggambarkan alur kerja suatu sistem informasi. Sebuah diagram aktivitas menunjukkan suatu alur kegiatan secara berurutan. Tabel 2.2 dibawah ini adalah simbol-simbol yang ada pada diagram aktifitas:

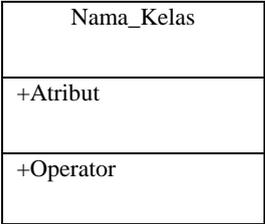
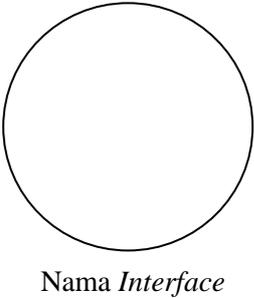
Tabel 2.2 Simbol-Simbol *Activity Diagram*.

Simbol	Deskripsi
Status awal 	Setatus awal aktivitas sistem, sebuah diagram aktivitas memiliki status awal
Aktivitas 	Aktivitas yang dilakukan sistem. Aktivitas biasanya diawali dengan kata kerja.
Pencabangan / <i>decision</i> 	Asosiasi penggabungan dimana lebih satu aktivitas.
<i>Fork</i> 	Digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel.
Penggabungan / <i>Join</i> 	Digunakan untuk menunjukkan kegiatan yang digabungkan.
<i>End Point</i> 	Mengakhiri aktivitas sistem.

2.2.6 Class Diagram

Diagram kelas menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki atribut yaitu variabel-variabel yang dimiliki oleh suatu kelas dan operasi atau metode yaitu fungsi-fungsi yang dimiliki oleh suatu kelas. Tabel 2.3 dibawah ini adalah simbol-simbol yang ada pada *clas diagram*:

Tabel 2.3 Bagan *Class Diagram*.

Simbol	Deskripsi
<p>Kelas</p> 	Kelas pada Struktur
<p>Interface</p> 	Metode pada <i>interface</i> yang digunakan pada suatu kelas sama persis dengan yang ada pada <i>interface</i> .
<p>Asosiasi</p> 	Relasi antara kelas dengan makna umum.
<p>Asosiasi Berarah</p> 	Relasi antar kelas dengan makna kelas yang satu digunakan pada kelas lain.

2.3 Penelitian Terdahulu

Tabel 2.4 berikut ini adalah beberapa penelitian terdahulu yang berkaitan dengan media pembelajaran :

Tabel 2.4 Penelitian Terdahulu

Nama	Judul	Terbit	Uraian
Foloruso, I. O., Abikoye, O. C., Jimoh, R. G., dan Raji. K. S.	A Rule Based Expert System For Mineral Identification	Journal Of Emerging Trends In Computing And Information Sciences, Vol 3, No 2, Februari 2012	Sistem ini dibangun dengan menggunakan metodologi <i>Forward Chaining</i> . Sistem ini digunakan untuk mengidentifikasi mineral tertentu dengan namanya. Sistem ini terdiri dari basis pengetahuan dasar yang merupakan database dalam sistem dan penalaran atau mesin inferensi. Database pada sistem ini menggunakan <i>microsoft access</i> .

Pada penelitian terdahulu sistem yang dibangun menggunakan metode *forward chaining* dan database yang digunakan adalah *microsoft access*. Kondisi yang dihasilkan adalah kondisi yang mendekati beberapa syarat tertentu kemudian akan dilakukan evaluasi jika ada perubahan terhadap hasil yang diinginkan.

Tabel 2.5 Penelitian Terdahulu Ke-2

Nama	Judul	Terbit	Uraian
Priti Srinivas Sajja, Dipti M Shah.	Knowledge Based Diagnosis Of Abdomen Pain Using Fuzzy Prolog Rules.	Department Of Computer Science & Technology Sardar Patel University, Vallabh Vidyanagar Gujarat, India. Vol. 1 No. 2. October 2010.	Dukungan teknologi informasi digunakan untuk pengambilan keputusan dalam ilmu kedokteran. Ledakan informasi di bidang medis membutuhkan sistem pendukung keputusan yang berorientasi pengetahuan untuk diagnosis sakit perut. Sistem basis pengetahuan terdiri dari beberapa aturan fuzzy yang mewakili pengetahuan domain dalam bentuk prolog sistem ini menggunakan prototype.

Pada penelitian terdahulu metode pengembangan perangkat menggunakan metode *Prototype* dan memiliki beberapa aturan *fuzzy* yang mewakili pengetahuan domain. Berbagai jenis heuristik yang disesuaikan dapat digabungkan dengan

struktur aturan prolog *fuzzy* yang digunakan dan bisa digunakan sebagai alat pelatihan dan dokumentasi.

Tabel 2.6 Penelitian Terdahulu Ke-3

Nama	Judul	Terbit	Uraian
Septilia Arfida.	Peranan Domain Penafsiran Dalam Menentukan Jenis Kuantor.	Jurnal Informatika Vol. 11 No. 1 Juni 2011	Logika sebagai salah satu bidang ilmu yang bisa berdiri sendiri. Dapat digunakan untuk mengevaluasi dan mengelompokan struktur dari argumen dan pernyataan – pernyataan yang diperoleh dari studi tentang pengaruh formal sistem dan argumen pada bahasa manusia sehari – hari. Kuantor merupakan kalimat ekspresi yang memuat kuantitas obyek yang terlibat misalnya ada semua, beberapa, tidak semua dan

Tabel 2.6 Penelitian Terdahulu Ke-3 (Lanjutan)

			lain – lain. Kuantor terdiri dari dua bentuk yaitu kuantor Universal dan Kuantor Eksistensial.
--	--	--	--

Pada penelitian terdahulu metode yang digunakan adalah kuantor universal dan kuantor eksistensial menggunakan logika *if – then, if –only if, and, or, dan not* sebagai penyelesaian pencarian nilai kebenaran. Kuantor merupakan kalimat ekspresi yang memuat kuantitas obyek yang terlibat misalnya ada semua, beberapa, tidak semua dan lain-lain. Kuantor terdiri dari dua bentuk yaitu kuantor universal dan kuantor eksistensial.

Tabel 2.7 Penelitian Terdahulu Ke-4

Nama	Judul	Terbit	Uraian
Yulmaini Muhamad Arifin	Perancangan Sistem Pendeteksi Pencemaran Air Menggunakan Casa Base Reasoning.	Jurnal Prosiding ICI-GAB 2012 Universitas IIB DARMAJAYA Vol. 1, No. 1. 2012 : ICI GAB 2012	Case Base Reasoning (CBR) merupakan salah satu bagian program dalam memahami kasus komputer berbasis, tujuan sistem adalah untuk menyelesaikan kasus baru dengan cara adaptasi

			<p>solusi pada kasus baru dengan cara adaptasi solusi pada kasus lama yang similiaritas dengan kasus baru. CBR dapat diaplikasikan dibidang Varous, seperti di bagian kesehatan dalam hal ini adalah sistem deteksi pencemaran air, sistem deteksi ini digunakan untuk mendeteksi pencemaran air permukaan dan sistem ini diharapkan dapat membantu memberikan informasi tentang tarikan air yang terjadi berdasarkan Indikator fisik dan juga sistem akan memberikan informasi pencegahan.</p>
--	--	--	---

Pada penelitian terdahulu metode yang digunakan adalah Observasi dan Wawancara, tujuan sistem adalah untuk menyelesaikan kasus baru dengan cara adaptasi solusi pada kasus lama yang similitas dengan kasus baru. Sistem deteksi ini digunakan untuk mendeteksi pencemaran air permukaan dan sistem ini memberikan informasi tentang tarikan air yang terjadi berdasarkan indikator fisik dan juga sistem akan memberikan informasi pencegahan.

Tabel 2.8 Penelitian Terdahulu Ke-5

Nama	Judul	Terbit	Uraian
Wahyu Ardianto, Wiwik Anggraeni, Ahmad Mukhlason	Pembuatan Sistem Pakar Untuk Pendeteksian Dan Penanganan Dini Pada Penyakit Sapi Berbasis Mobile Android Dengan Kajian Kinerja Teknik Knowledge Representasion	Jurnal Teknik ITS Vol. 1 September @012 ISSN: 2301 - 9271	<p><i>Knowledge Representation</i> adalah suatu proses untuk menangkap sifat-sifat penting pada sebuah permasalahan dan membuat informasi tersebut dapat diakses oleh prosedur pemecahan</p> <p><i>Knowledge Representation</i> merupakan bidang dari <i>Artificial Intelligence</i> atau kecerdasan buatan yang bersangkutan dengan bagaimana sebuah pengetahuan direpresentasikan</p>

Tabel 2.8 Penelitian Terdahulu Ke-5 (Lanjutan)

			secara simbolis dan dimanipulasi secara otomatis oleh program penalaran
--	--	--	---

Pada penelitian terdahulu metode yang digunakan adalah *Knowledge Representation*, *Frame Based Knowledge Representation*, dan *Rule Based Representation*. Hasil yang dikemukakan tidak terlepas dari fakta dan data – data sebelumnya sebagai bahan pertimbangan.

2.4 Android

2.4.1 Sejarah *Android*

Menurut Kasman (2013, p.2) menguraikan bahwa “*Android* merupakan sebuah sistem operasi yang berbasis *Linux* untuk perangkat *portable* seperti *smartphone* dan komputer tablet”. *Android* menyediakan *platform* terbuka bagi programmer untuk mengembangkan aplikasi sendiri pada berbagai perangkat dengan sistem operasi *android*.

Menurut Yuni Puspita Sari (Jurnal Informatika, Vol. 16, No. 1, Juni 2016) *Android* adalah sebuah sistem operasi untuk *smartphone* dan Tablet. Sistem operasi dapat diilustrasikan sebagai ‘jembatan’ antara piranti (*device*) dan penggunaanya, sehingga pengguna bisa berinteraksi dengan *device*-nya dan menjalankan aplikasi-aplikasi yang tersedia pada *device*.

Menurut Evi Yulianti (SEMBISTEK 2014 IBI DARMAJAYA ISSN : 2407-6171) *Android* merupakan software stack untuk perangkat mobile yang meliputi sistem operasi, middleware, dan aplikasi. *Android* merupakan tumpukan software, karena komponen-komponen pada *Android* dirancang membentuk tumpukan

Android merupakan sistem operasi untuk telepon seluler berbasis *linux* sebagai kernelnya. *Android* menyediakan platform terbuka (*open source*) bagi para pengembang untuk menciptakan aplikasi mereka sendiri. Awalnya, perusahaan *search engine* terbesar saat ini, yaitu *Google Inc*, membeli *Android Inc*, pendatang baru yang membuat perangkat lunak untuk ponsel. *Android Inc*. Didirikan oleh Andy Rubin, Rich Milner, Nick Sears dan Chris White pada tahun 2003. Pada Agustus 2005 *Google* membeli *Android Inc*. Dimulai pada tahun 2005, *Android Inc*. dibawah naungan *Google Inc*. Berusaha membuat sebuah operating system mobile baru. Sejak saat itulah mulai beredar rumor bahwa *Google* akan melakukan ekspansi bisnis ke industri seluler. Akhirnya pada bulan September 2007 *Google* mengajukan hak paten atas produknya yang dinamai *Nexus One*.

Kemudian pada akhir tahun 2008, dibentuk sebuah tim kerja sama yang dinamai *Open Handset Alliance (OHA)*. *OHA* ini terdiri dari beberapa produsen perangkat telekomunikasi ternama dunia, antara lain *ASUS*, *Toshiba*, *Sony Ericsson* (sekarang *Sony*), *Garmin*, *Vodafone*, dan *Softbank*. *OHA* bekerja sama untuk mengembangkan sebuah kernel *Linux* yang akan dijadikan sebuah program untuk perangkat seluler. Hingga akhirnya *OHA* berhasil dan mengumumkan produk operating system mobile yang diberi nama *Android*. Ponsel yang mendapat kehormatan untuk mencoba pertama kali sistem operasi *Android* adalah *HTC Dream*.

HTC Dream dirilis pada bulan Oktober tahun 2008. Sejak saat itu banyak perusahaan perangkat seluler di dunia ikut menggunakan *Android* sebagai operating system ponsel mereka.

2.4.2 Komponen *Android*

Android SDK (Software Development Kit) merupakan sebuah *tools* yang diperlukan untuk mengembangkan aplikasi berbasis *Android* menggunakan bahasa pemrograman *Java*. Pada saat ini *Android SDK* telah menjadi alat bantu dan *API (Application Programming Interface)* untuk mengembangkan aplikasi berbasis *android*. *Android SDK* dapat anda lihat dan unduh pada situs

resminya, yaitu www.developer.android.com, *Android* SDK bersifat gratis dan bebas anda distribusikan karena *Android* bersifat *open source*.

2.4.3 Kelebihan Sistem Operasi *Android*

Kelebihan dari sistem operasi *Android* adalah sebagai berikut :

1) *Complete Platform*

Sistem operasi *Android* adalah sistem operasi yang banyak menyediakan *tools* yang berguna untuk membangun sebuah aplikasi yang kemudian aplikasi tersebut dapat lebih dikembangkan lagi oleh para *developer*.

2) *Open Source Platform*

Platform Android yang bersifat *open source* menjadikan sistem operasi ini mudah dikembangkan oleh para *developer* karena bersifat terbuka.

3) *Free Platform*

Developer dengan bebas bisa mengembangkan, mendistribusikan dan memperdagangkan sistem operasi *Android* tanpa harus membayar royalti untuk mendapatkan *license*.

2.4.4 Versi *Android*

Kasman (2013, p.3) menyatakan bahwa versi-versi *android* yang telah dirilis adalah sebagai berikut :

- 1) *Angel Cake* (1.0) dirilis tanggal 28 September 2008
- 2) *Battenberg* (1.1) dirilis tanggal 9 Februari 2009
- 3) *Cupcake* (1.5) dirilis tanggal 30 April 2009
- 4) *Donut* (1.5) dirilis tanggal 15 September 2009
- 5) *Eclair* (2.0-2.1) dirilis tanggal 26 Oktober 2009

- 6) *Froyo* (2.2) dirilis tanggal 20 Mei 2010
- 7) *Gingerbread* (2.3) dirilis tanggal 6 Desember 2010
- 8) *Honeycomb* (3.1) dirilis tanggal 10 Mei 2011
- 9) *Ice Cream Sandwich* dirilis tanggal 16 Desember 2011
- 10) *Jelly Bean* (4.1) dirilis tanggal 9 Juli 2011
- 11) *Kitkat* (4.4) dirilis tanggal 31 Oktober 2013
- 12) *Lollipop* (5.0) dirilis tanggal 15 Oktober 2014



Gambar 2.1 Evolusi *Android*.

2.5 Perangkat Lunak yang Digunakan

2.5.1 *Adobe Flash Professional CS 6*

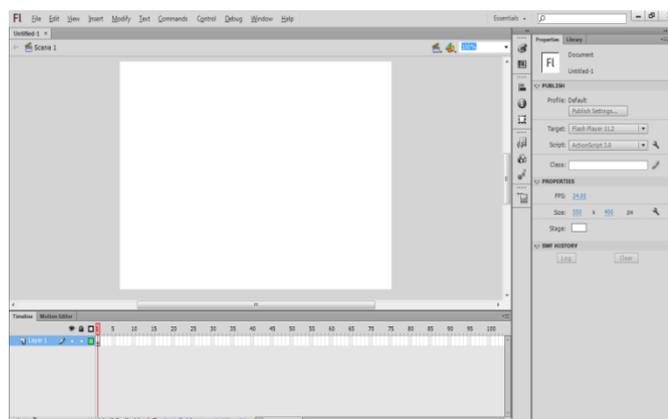
Menurut Madcoms (2012, p.2) *Adobe Flash Professional CS6* merupakan versi terbaru dari *Adobe Flash* yang dirilis tahun 2012 sebagai penyempurna dari versi *adobe flash* sebelumnya, yaitu *adobe flash CS5* dimana pada versi ini telah mampu mengolah *teks* maupun objek dengan efek 3 dimensi sehingga tampak lebih menarik.

Hasil penyimpanan *file* dari *Adobe Flash Professional CS6* tidak dapat dibuka pada program *flash* versi sebelumnya. Apabila hasil *file* ingin dibuka pada versi sebelumnya, maka pada format penyimpanannya pilih tipe *flash* pada bagian *save as a type*. Area kerja *Adobe Flash Professional CS6* dirancang secara khusus agar ruang kerja yang digunakan dapat diatur dan lebih mudah dipahami oleh pemakai pemula maupun para desainer flash yang telah berpengalaman. *Adobe Flash Professional CS6* adalah program yang cukup

kompleks sehingga tidak mungkin untuk menjabarkannya secara lengkap. Disini hanya akan diberikan sedikit pengertian dan fungsi dasar dari *Adobe Flash Professional CS6*.

2.5.2 User Interface

Tampilan awal dalam membuka program *Adobe Flash Professional CS6* akan muncul layar seperti pada gambar 2.2 berikut :

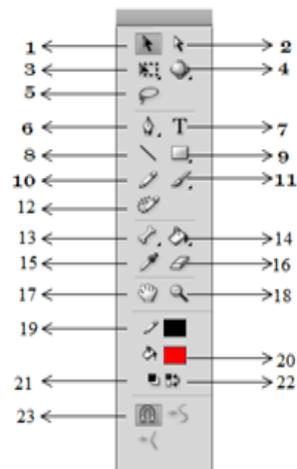


Gambar 2.2 User Interface Adobe Flash CS6.

Tampilan pada gambar 2.2 merupakan *interface* dimana kita akan memulai membuat sebuah *flash*. Tampilan putih ditengah adalah merupakan *stage* tempat kita membuat tampilan. Pada bagian kanan adalah *toolbox* berfungsi untuk membuat sesuatu pada tampilan. Sedangkan pada bagian bawah merupakan *timeline*.

2.5.3 Toolbox

Toolbox adalah sebuah panel yang menampung tombol-tombol yang berguna untuk membuat suatu desain animasi mulai dari tombol *selection*, *subselection*, *free transform*, *3D rotation*, *lasso*, *pen*, *text*, *line*, *rectangle*, *pencil*, *brush*, *deco*, *bone*, *paint bucket*, *eyedropper*, *eraser*, *hand*, *zoom*, *stroke color*, *fill color*, *black and white*, *swap color*, *snap to object*, dan *option* pendukung dari tombol yang terpilih seperti pada gambar 2.3 berikut :



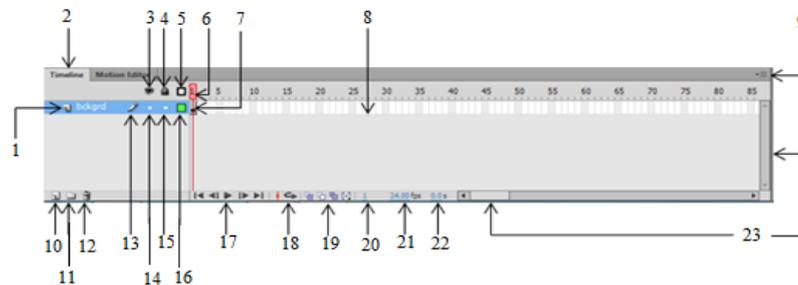
Gambar 2.3 Toolbox Adobe Flash Professional CS6.

- 1) *Selection Tool (V)* berfungsi untuk menyeleksi objek.
- 2) *Subselection Tool (A)* berfungsi untuk menyeleksi bagian objek untuk proses editing.
- 3) *Free Transform Tool (Q)* berfungsi untuk mengubah bentuk objek secara bebas.
- 4) *3D Rotation Tool (W)* berfungsi untuk melakukan rotasi 3D pada objek berdasarkan sumbu X, Y, dan Z.
- 5) *Lasso Tool (L)* berfungsi untuk menyeleksi objek dengan pola seleksi bebas.
- 6) *Pen Tool (P)* berfungsi untuk menggambar objek.
- 7) *Text Tool (T)* berfungsi untuk mengetik teks dan paragraf.
- 8) *Line Tool (N)* berfungsi untuk menggambar objek garis lurus.
- 9) *Rectangle Tool (R)* berfungsi untuk menggambar objek kotak.
- 10) *Pencil Tool (Y)* berfungsi untuk menggambar dengan bentuk goresan pensil.
- 11) *Brush Tool (B)* berfungsi untuk menggambar dengan polesan kuas.
- 12) *Deco Tool (U)* berfungsi untuk menggambar corak dekorasi dengan menggunakan simbol grafik
- 13) *Bone Tool (X)* berfungsi untuk membuat animasi pertulangan dengan menambahkan titik sendi pada objek.
- 14) *Paint Bucket Tool (K)* berfungsi untuk memberi warna bidang objek.

- 15) *Eyedropper Tool(I)* berfungsi untuk mengambil sampel warna dari sebuah object.
- 16) *Erasser Tool(E)* berfungsi untuk menghapus bidang object.
- 17) *Hand Tool(H)* berfungsi untuk menggeser area lembar kerja atau stage.
- 18) *Zoom Tool(M, Z)* berfungsi untuk memperbesar atau memperkecil tampilan lembar kerja atau stage.
- 19) *Stroke Color* berfungsi untuk menentukan warna garis.
- 20) *Fill Color* berfungsi untuk menentukan warna bidang object.
- 21) *Black and White* berfungsi untuk mengubah warna garis dan bidang menjadi hitam dan putih.
- 22) *Swap Color* berfungsi untuk membalikan warna antara warna garis dan warna bidang object.
- 23) *Snap to Object* berfungsi untuk mengaktifkan atau mematikan fungsi
Snap to Object.

2.5.4 Timeline

Timeline berfungsi untuk menentukan jumlah layer, durasi animasi, *frame*, menempatkan script dan beberapa keperluan animasi lainnya. Semua bentuk animasi yang anda buat akan diatur dan ditempatkan pada layer dalam *timeline*, seperti pada gambar 2.4 berikut :



Gambar 2.4 *Timeline Adobe Flash Professoanl CS6.*

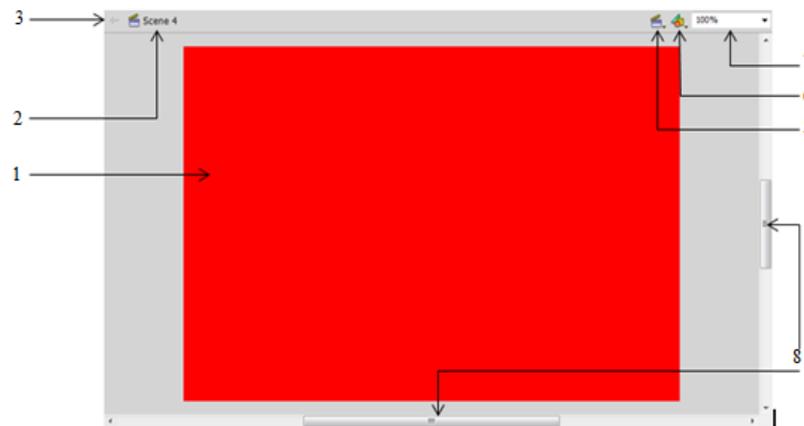
- 1) *Layer*, lembar kerja yang menampung objek yang akan dianimasikan didalam timeline.
- 2) *Timeline*, tabulasi dari lembar kerja atau stage yang sedang dikerjakan.

- 3) *ShoworHideAllLayers*, untuk menyembunyikan atau menampilkan semua isi layer.
- 4) *LockorUnlockAllLayers*, untuk mengunci atau melepas kunci objek dari semua layer.
- 5) *ShowAllLayerasOutlines*, untuk menampilkan objek pada semua layer dalam bentuk *outline*.
- 6) *Playhead*, jarum untuk membaca *frame* pada saat animasi dijalankan.
- 7) *BlankKeyframe*, sebuah simbol lingkaran kosong yang menampung suatu objek.
- 8) *Frame*, suatu bagian dari *layer* yang digunakan untuk mengatur pembuatan animasi.
- 9) Tombol *Menu*, untuk mengatur tampilan *frame*.
- 10) *NewLayer*, untuk menambah *layer* baru.
- 11) *NewFolder*, untuk menambah *folder* baru.
- 12) *Delete*, untuk menghapus *layer*.
- 13) Simbol Pensil, menunjukkan bahwa *layer* dalam kondisi terpilih atau aktif.
- 14) Titik *Show* or *Hide*, klik untuk menampilkan atau menyembunyikan *layer* aktif.
- 15) Titik kunci, klik untuk mengunci atau melepas kunci *layer* yang aktif.
- 16) Kotak *Outline*, kilik untuk menampilkan objek dalam layer aktif menjadi bentuk *outline*.
- 17) *Controler*, tombol yang digunakan untuk mengontrol animasi.
- 18) *Loop*, tombol yang digunakan untuk mengaktifkan pengulangan animasi.
- 19) Tombol Pengatur tampilan animasi, untuk mengatur tampilan animasi didalam *stage*.
- 20) *CurrentFrame*, menunjukkan posisi *frame* aktif.
- 21) *FrameRate*, untuk mengatur kecepatan gerak animasi dalam tiap detiknya.

- 22) *ElapsedTime*, menunjukkan durasi atau lamanya animasi.
- 23) *Scrollbar*, menggulung jendela *timeline* secara vertikal atau horizontal.

2.5.5 Stage

Stage adalah lembar kerja yang digunakan untuk membuat atau mendesain objek yang akan dianimasikan. Objek yang dibuat dalam lembar kerja dapat berupa objek *Vektor*, *Movieclip*, *Text*, *Button*, dan lain-lain. Seperti pada Gambar 2.5 berikut :



Gambar 2.5 Stage Adobe Flash Professional CS6.

- 1) *Stage*, lembar kerja untuk menyusun objek yang akan dianimasikan.
- 2) *Scene*, menunjukkan nama *scene* yang aktif.
- 3) Panah yang digunakan untuk berpindah dari lembar kerja simbol ke lembar kerja utama.
- 4) *EditScene*, untuk memilih nama *scene* yang akan diedit.
- 5) *EditSymbols*, untuk memilih nama simbol yang akan diedit.
- 6) *Zoom*, untuk mengatur besarnya tampilan stage atau lembar kerja.
- Scrollbar*, untuk menggulung lembar kerja secara horisontal dan vertikal.

2.5.6 Action Script 3.0

Madcoms (2012, p.5) mendefinisikan *Action Script* adalah bahasa pemrograman untuk *Adobe Flash Player* dan *Adobe AIR Environment*. *ActionScript* dijalankan oleh *Action Script Virtual Machine* yang merupakan bagian dari *Flash Player* dan *AIR*. Koding *Action Script* biasanya dikompilasi ke dalam format *bytecode* (bahasa pemrograman yang ditulis dan dipahami oleh komputer) oleh *compiler*, seperti yang dibangun ke dalam *Adobe Flash CS6 Professional* atau *Adobe Flash Builder* dan yang tersedia di dalam *Adobe Flex SDK* dan *Flex Data Services*. *Bytecode* ini tertanam dalam file *SWF*, yang dijalankan oleh *Flash player* maupun *AIR*

2.5.7 APK

Madcoms (2012, p.10) menjelaskan bahwa *Application Package File* atau berkas paket aplikasi *Android* (APK) adalah format berkas yang digunakan untuk mendistribusikan dan memasang *software* ke *ponsel* dengan sistem operasi *Android*, mirip dengan paket *MSI* pada *Windows* atau *Deb* pada *OS Debian*.

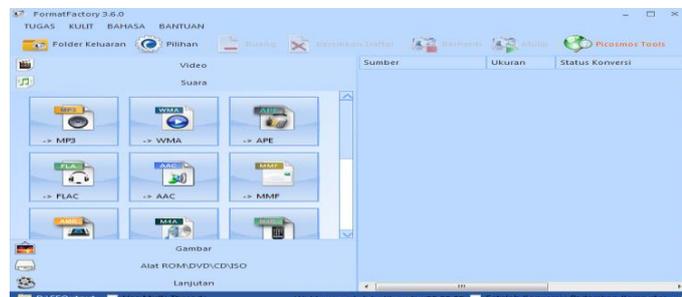
2.5.8 Format Factory

Format Factory adalah aplikasi *converter file* yang serbaguna. Sebab *software* ini bisa melakukan berbagai tugas mengubah *format file* secara baik dan benar. Bisa untuk *convert* video, bisa untuk *convert* gambar, *audio*, *format DVD*, dan masih ada beberapa tugas lainnya yang bisa dilakukan oleh *format factory*.

Ada beberapa nilai tambah dari aplikasi ini dibanding aplikasi sejenis lainnya:

- 1) Gratis, *format factory* ini gratis.
- 2) Fitur lengkap, seperti yang sudah dijelaskan, fiturnya sangat lengkap dan begitu bermanfaat bagi kita.
- 3) *User Friendly*, apalah arti sebuah *software* super namun sulit digunakan, berbeda dengan *format factory* yang super dan mudah digunakan.

Berikut ini adalah tampilan awal dari *format factory* yang bisa dilihat pada gambar 2.6 berikut:



Gambar 2.6 User Interface Format Factory.

2.6 Sistem Pakar.

Sistem pakar (*expert system*) adalah sistem yang berusaha mengadopsi pengetahuan manusia ke komputer yang dirancang untuk menyelesaikan masalah seperti layaknya seorang pakar.

Menurut Turban (2005), keahlian dipindahkan dari pakar ke suatu komputer. Pengetahuan ini kemudian disimpan di dalam komputer. Pada saat pengguna menjalankan komputer untuk mendapatkan informasi, sistem pakar menanyakan fakta-fakta dan dapat membuat penalaran (inferensi) dan sampai pada suatu kesimpulan. Kemudian, sistem pakar memberikan penjelasan (memberikan kesimpulan atas hasil konsultasi yang telah dilakukan sebelumnya).

Dengan sistem pakar ini, orang awam pun dapat menyelesaikan masalahnya atau hanya sekedar mencari suatu informasi berkualitas yang sebenarnya hanya dapat diperoleh dengan bantuan para ahli dibidangnya. Sistem pakar ini juga dapat membantu aktivitas para pakar sebagai asisten yang mempunyai pengetahuan yang dibutuhkan.

2.6.1 Ciri-Ciri Sistem Pakar.

Ciri-ciri sistem pakar menurut Sutojo (2011) dalam Pratama (2015), adalah:

1. Terbatas pada domain keahlian tertentu.
2. Dapat memberikan penalaran-penalaran untuk data-data yang tidak lengkap atau tidak pasti.

3. Padat penebaran optimal.
4. Dirancang untuk dapat dikembangkan secara bertahap.
5. Pengetahuan dan mekanisme penalaran (inference) jelas terpisah.
6. Keluarannya bersifat anjuran.
7. Sistem dapat mengaktifkan kaidah secara searah yang sesuai dituntun oleh dialog dengan user.
8. Sistem dapat mengaktifkan kaidah secara searah yang sesuai dituntun oleh dialog dengan user.

2.6.2 Manfaat Sistem Pakar

Sistem pakar menjadi populer karena sangat banyak kemampuan dan manfaat yang diberikannya, diantaranya:

1. Meningkatkan produktivitas, karena sistem pakar dapat bekerja lebih cepat daripada manusia.
2. Membuat seorang yang awam bekerja seperti layaknya seorang pakar.
3. Meningkatkan kualitas dengan memberi nasihat yang konsisten dan mengurangi kesalahan.
4. Mampu menangkap pengetahuan dan kepakaran seseorang.
5. Dapat beroperasi di lingkungan yang berbahaya.
6. Memudahkan akses pengetahuan seorang pakar.
7. Handal, sistem pakar tidak pernah menjadi bosan dan kelelahan atau sakit.
8. Meningkatkan kapabilitas sistem komputer. Integrasi sistem pakar dengan sistem komputer lain membuat sistem lebih efektif dan mencakup lebih banyak aplikasi.
9. Mampu bekerja dengan informasi yang tidak lengkap atau tidak pasti.
10. Bisa digunakan sebagai media pelengkap dalam pelatihan. Pengguna pemula yang bekerja dengan sistem pakar akan menjadi lebih berpengalaman karena adanya fasilitas penjelas yang berfungsi sebagai guru.
11. Meningkatkan kemampuan untuk menyelesaikan masalah karena sistem pakar mengambil sumber pengetahuan dari banyak pakar.

2.7 Rule Based Knowledge

Knowledge base memuat fakta-fakta yang menjelaskan area masalah, dan juga teknik menerangkan masalah yang menjelaskan bagaimana fakta-fakta tersebut cocok satu dengan yang lain dalam urutan yang logis. Atau dapat dikatakan *knowledge base* terdiri dari fakta yang menggambarkan area problem atau problem domain dan juga teknik penyajian yang menggunakan fakta sesuai logika. Istilah problem domain digunakan untuk menjelaskan area masalah. Teknik menerangkan masalah (*knowledge representation technique*) yang populer adalah penggunaan aturan. Rule/aturan merupakan rincian dalam situasi yang tidak berubah. Aturan menentukan apa yang harus dilakukan dalam situasi tertentu dan terdiri dari dua bagian :

1. Kondisi benar dan tidak benar.
2. Tindakan yang diambil bila kondisi benar.

Semua aturan yang ada dalam sistem pakar disebut perangkat aturan (rule set). Perangkat aturan dapat bervariasi dari sekitar selusin aturan untuk sistem pakar yang sederhana sampai 500, 1000, atau 10.000 aturan untuk sistem pakar yang rumit.

2.8 Ikan Air Tawar

Ikan membutuhkan adaptasi fisiologis yang bertujuan menjaga keseimbangan konsentrasi ion dalam tubuh, 41% dari seluruh spesies ikan diketahui berada di air tawar. Insang mereka harus mampu mendifusikan air sembari menjaga kadar garam dalam cairan tubuh secara simultan. Ikan air tawar memiliki beberapa kendala dalam pengembangbiakan salah satunya seperti penyakit yang mungkin ditimbulkan oleh faktor virus dan lingkungan. Pada umumnya penyakit yang diderita air tawar ada 2 yaitu penyakit menular dan tidak menular. Penyakit menular ikan air tawar antara lain seperti parasit, jamur, bakteri, virus sedangkan penyakit yang tidak menular adalah penyakit akibat lingkungan, penyakit malnutrisi, dan penyakit genetik. Usaha perikanan air tawar pada kolam banyak terdapat dalam masyarakat kita terutama di daerah - daerah yang jauh dari laut, usaha ini dilakukan sebagai usaha sampingan maupun utama yang dapat diusahakan

dipekarangan rumah atau lahan yang kosong. Tempat pemeliharaan ikan ini dinamakan bermacam – macam diantaranya ada yang dinamakan tambak, siwakan, Kolam, Keramba dan lain – lain.

2.8.1 Jenis – Jenis Yang Dipelihara Pada Air Tawar

Pemeliharaan ikan air tawar pada umumnya dilakukan hanya untuk memelihara satu jenis ikan saja, tetapi ada juga yang dilakukan pemeliharaannya secara campuran hal ini disebabkan didalam kolam sebenarnya sudah terdapat dengan sendirinya berbagai macam makanan untuk berbagai jenis ikan, walaupun demikian kita harus memperhatikan jenis ikan apa yang cocok sebagai peliharaan pokok dan peliharaan tambahan. Oleh sebab itu percampuran jenis – jenis ikan itu harus sangat diperhatikan sehingga ikan – ikan tersebut tidak saling dirugikan. Adapun ikan – ikan yang dapat dilakukan pemeliharaannya pada air tawar adalah sebagai berikut :

1. Ikan Mujaer
2. Ikan Tawes
3. Ikan Mas
4. Ikan Nila
5. Ikan Bandeng
6. Ikan Sepat Siam
7. Ikan Lele
8. Ikan Patin
9. Ikan Gurame

2.9 Penyakit Ikan Air Tawar

Kegiatan budidaya ikan air tawar konsumsi merupakan kegiatan yang mempunyai resiko yang tinggi karena ikan merupakan makhluk bernyawa yang kapan saja mengalami kematian. Salah satu penyebab gagalnya kegiatan budidaya ikan adalah karena faktor penyakit. Munculnya penyakit pada ikan umumnya merupakan hasil interaksi kompleks/ tidak seimbang antara tiga komponen dalam

ekosistem perairan yaitu inang (ikan yang lemah), patogen yang ganas, dan kualitas lingkungan yang memburuk (Afrianto).

Secara umum penyakit ikan dapat dikelompokkan menjadi dua, yaitu:

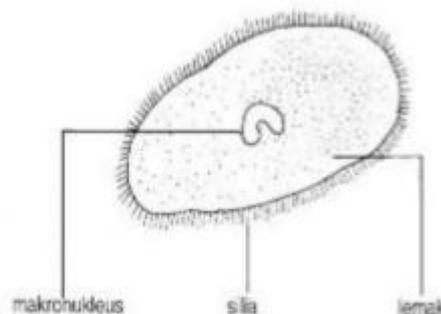
1. Penyakit infeksi

Penyakit infeksi merupakan penyakit yang disebabkan oleh:

A. Parasit

Akibat dari penyakit yang disebabkan oleh parasit secara ekonomis cukup merugikan yaitu dapat menyebabkan kematian, menurunkan bobot, bentuk serta ketahanan tubuh ikan sehingga dapat dimanfaatkan sebagai jalan masuk bagi infeksi sekunder oleh patogen lain seperti jamur, bakteri dan virus. Salah satu jenis *protozoa* yang paling sering menjadi kendala dalam budidaya ikan adalah *Ichthyophthirius multifiliis* atau *ich* (penyakit bintik putih).

Sifat serangannya sangat sporadis dan kematian yang diakibatkannya dapat mencapai 100% populasi dalam tempo yang relatif singkat. Secara umum gejala ikan yang terserang protozoa adalah ikan tampak pucat, nafsu makan berkurang, gerakan lambat (sering menggosok-gosokkan tubuhnya pada dinding kolam), adanya bercak-bercak putih pada permukaan tubuh ikan, dan pada infeksi lanjut ikan megap-megap dan meloncat-loncat ke permukaan air untuk mengambil oksigen.



Gambar 2.7. *Ichthyophthirius multifiliis* (Afrianto, 1992)

Secara umum gejala ikan yang terserang protozoa adalah :

1. Ikan tampak pucat.
2. Nafsu makan berkurang.
3. Adanya bercak putih pada tubuh
4. Gerakan lambat.
5. Menggosokkan tubuh ke di dinding kolam.
6. Ikan mangap-mangap.
7. Loncat ke permukaan air mengambil oksigen.

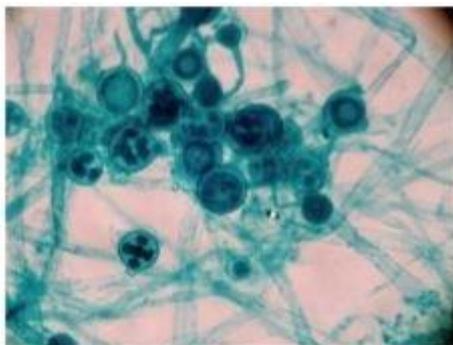
Parasit kedua yaitu metazoa, golongan metozoa antara lain *Monogenetic trematod* (golongan cacing), *cestoda*, *nematoda*, *Cepopoda* (*Argulus sp*, *Lernaea sp* dan golongan *Isopoda*). organ yang menjadi target serangan parasit ini adalah insang. penularan terjadi secara horisontal terutama pada saat cacing dalam fase berenang bebas yang sangat infeksiif. secara umum gejala dari serangan *metazoa* adalah :

1. Ikan tampak lemah.
2. Tidak nafsu makan.
3. Pertumbuhan lambat.
4. Berenang tidak normal.
5. Produksi lendir berlebihan.
6. Ikan berkumpul disekitar air masuk (kadar oksigen tinggi).
7. Insang pucat dan membengkak.
8. Ikan sulit bernafas (kekurangan oksigen).
9. Badan kemerahan disekitar parasit yang menempel
10. Infeksi yang berat mengakibatkan parasit terlihat jelas.

B. Jamur

Semua jenis ikan air tawar termasuk telurnya rentan terhadap infeksi jamur. Beberapa faktor yang sering memicu terjadinya infeksi jamur adalah penanganan yang kurang baik (transportasi) sehingga menimbulkan luka pada tubuh ikan, kekurangan gizi, suhu, dan oksigen terlarut yang rendah, bahan

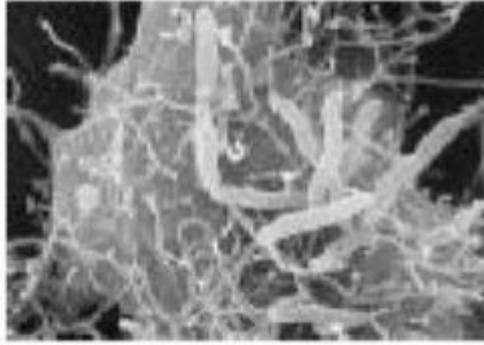
organik tinggi, kualitas telur buruk/tidak terbuahi dan padatnya telur pada jaring yang disediakan untuk pembudidayaan telur ikan. Penyakit ini menular terutama melalui spora di air. Gejala-gejalanya dapat dilihat secara klinis adanya benang-benang halus menyerupai kapas yang menempel pada telur. Salah satu jenis jamur yang sering menjadi kendala adalah dari family saprolegniaceae.



Gambar 2.8. Saprolegniaceae (Afrianto, 1992)

C. Bakteri

Penyakit yang disebabkan oleh bakteri adalah penyakit yang paling banyak menyebabkan kegagalan pada budidaya ikan air tawar. Penyakit akibat infeksi bakterial masih sering terjadi dengan intensitas yang variatif. Umumnya pembudidaya masih mengandalkan antibiotik sebagai “magic bullet” untuk melawan penyakit bakterial. Jenis penyakit yang disebabkan oleh bakteri antara lain adalah penyakit merah yang disebabkan oleh bakteri gram negatif (*Aeromonas hydrophila*), penyakit columnaris atau luka kulit, sirip dan inang yang disebabkan oleh infeksi bakteri *Flavobacterium columnare*, penyakit tuberculosis yang tergolong sangat kronis disebabkan oleh bakteri gram positif *Mycobacterium* sp, dan penyakit Streptocociasis yang disebabkan oleh bakteri gram positif *Streptococcus* sp. *Aeromonas* sp.



Gambar 2.9. *Aeromonas* sp (Afrianto, 1992)

D. Virus

Patogen virus juga menyebabkan penyakit pada budidaya ikan air tawar. Belum banyak diketahui penyakit yang disebabkan oleh virus di Indonesia kecuali penyakit Lymphocystis dan Koi Hervesvirus (KHV). Infeksi Lymphocystis hanya bersifat kronis dan bila menyerang ikan hias akan mengalami kerugian yang berarti merusak keindahan ikan. Sampai saat ini KHV merupakan penyakit yang paling serius dan sporadis terutama untuk komoditi ikan mas koi.

E. White Spot (Ich)

White Spot atau dikenal juga sebagai penyakit “ich” merupakan penyakit ikan yang disebabkan oleh parasit. Penyakit ini umum dijumpai pada hampir seluruh spesies ikan secara potensial *white spot* dapat berakibat mematikan. Penyakit ini ditandai dengan munculnya bintik-bintik putih disekujur tubuh dan juga sirip. siklus hidup dan cara memperbanyak diri inang *white spot* yang bervariasi memegang peranan penting terhadap berjangkitnya penyakit tersebut. siklus hidup *white spot* terdiri dari beberapa tahap, tahapan tersebut secara umum dapat dibagi dua yaitu tahapan infeksi dan tahapan tidak infeksi. gejala klinis *white spot* merupakan akibat dari bentuk tahapan siklus infeksi. wujud dari *white spot* pada tahapan infeksi ini dikenal sebagai *trophont*. *trophont* hidup dalam lapisan epidermis kulit insang atau rongga mulut. parasit ini hidup di lapisan dalam kulit berdekatan dengan lapisan basal lamina. ikan-ikan yang terjangkit akan menunjukkan penampakan

berupa bintik-bintik putih pada sirip, tubuh, insang atau mulut. pada awal perkembangannya bintik tersebut tidak akan bias dilihat dengan mata. tapi pada saat parasit tersebut makan tumbuh dan membesar sehingga bisa mencapai 0.5 – 1 mm bintik tersebut dapat dengan mudah dikenali.

F. Black Spot

Ikan yang menderita penyakit ini akan muncul titik hitam kecil pada tubuhnya secara umum penyakit ini mudah disembuhkan. sering kali penyakit *Black Spot* terjadi pada akuarium baru saat ikan-ikan dimasukkan. Semua ikan berisiko terkena penyakit ini tetapi ikan Piranha paling rentan.

2. Penyakit non-infeksi

Penyakit non-infeksi merupakan penyakit yang diakibatkan karena faktor:

A. Lingkungan

Pengaruh dari penyakit yang diakibatkan faktor lingkungan sering mengakibatkan kerugian yang serius karena kematian yang berlangsung sangat cepat, tiba-tiba, dan mematikan seluruh populasi ikan. penyebabnya seperti keracunan akibat peledakan populasi plankton, keracunan pestisida/limbah industri, bahan kimia, dan lainnya.

Faktor lingkungan yang buruk akan menyebabkan ikan menjadi:

1. Tercekik yaitu kekurangan oksigen yang umumnya terjadi menjelang pagi hari pada perairan yang punya populasi *Phytoplankton* tinggi.
2. Keracunan *Nitrit*, *Nitrit* (NO_2^-) adalah ion-ion anorganik alami, yang merupakan bagian dari siklus *nitrogen*. Aktifitas *mikroba* di tanah atau air menguraikan sampah yang mengandung *nitrogen* organik pertama-pertama menjadi *ammonia*, kemudian dioksidasikan menjadi *nitrit*. pengaruh dari *nitrit* menimbulkan penyakit darah cokelat karena disebabkan oleh konsentrasi *nitrit* yang tinggi di dalam air yang berasal dari hasil metabolisme ikan.

3. Keracunan *ammonia*, terjadi hampir sama dengan *nitrit* tetapi pada umumnya karena pengaruh pemberian pakan yang berlebihan atau bahan organik, sedangkan populasi bakteri pengurai tidak mencukupi. Yang sangat beracun adalah dalam bentuk NH_3 .

4. Fluktuasi suhu air yang ekstrim, dimana perubahan suhu air yang ekstrim akan merusak keseimbangan hormonal dan fisiologis tubuh ikan dan pada umumnya ikan tidak mampu untuk beradaptasi terhadap perubahan dan mengakibatkan ikan stress bahkan kematian.

B. Malnutrisi

Pemberian pakan yang berlebihan atau kekurangan dan tidak teratur juga dapat menyebabkan penyakit pada ikan. Penyakit karena malnutrisi jarang menunjukkan gejala spesifik sehingga agak sulit didiagnosa penyebab utamanya. Tetapi dalam diet pakan dapat mengakibatkan kelainan fungsi morfologis dan biologis seperti defisiensi asam *panththenic* penyakit jaring insang ikan yang dapat menyebabkan ikan sulit bernafas yang diikuti dengan kematian, defisiensi vitamin A yang menyebabkan mata menonjol / buta dan terjadi pendarahan pada kulit juga ginjal, defisiensi vitamin B-1 yang menyebabkan kehilangan nafsu makan, pendarahan, dan penyumbatan pembuluh darah, defisiensi asam lemak esensial yang berakibat infiltrasi lemak pada kulit dan minimnya pigmentasi pada tubuh ikan. Yang cukup berbahaya adalah karena defisiensi vitamin C yang merupakan penyakit yang umum terjadi dimana akibat yang paling populer adalah *broken back syndrome* seperti *scoliosis* dan *lordosis*.

C. Genetis

Salah satu penyebab penyakit yang kompleks pada kegiatan budidaya ikan air tawar karena adanya faktor genetik terutama karena adanya perkawinan satu keturunan (*inbreeding*). pernikahan *inbreeding* yang dilakukan secara terus menerus akan menurunkan kualitas ikan berupa variasi genetik dalam tubuh ikan.

Akibat dari pernikahan *inbreeding* adalah:

1. Pertumbuhan ikan lambat (bantet/kontet) dan ukuran beragam.
2. Lebih sensitif terhadap infeksi patogen.
3. Organ tubuh badan yang tidak sempurna serta kelainan lainnya.