

BAB 2

TINJAUAN PUSTAKA

2.1 Landasan Teori

Menurut Kamus Besar Bahasa Indonesia Landasan adalah dasar atau tumpuan, sedangkan Teori adalah pendapat yang didasarkan pada penelitian dan penemuan yang didukung oleh data dan argumentasi. Selain itu Teori juga berarti generalisasi atau kumpulan generalisasi yang dapat digunakan untuk menjelaskan berbagai fenomena secara sistematis. (William Wiersma dalam Sugiyono, 2010:52).

Suatu landasan teori dari suatu penelitian disini bisa disimpulkan sebagai studi literatur atau tinjauan pustaka. Hasil dari landasan teori atau kajian teori ini diperoleh kesimpulan-kesimpulan atau pendapat-pendapat para ahli, kemudian dirumuskan pada pendapat baru

2.1.1 Anak Berkebutuhan Khusus tunarungu (ABK tunarungu)

Secara umum anak tunarungu dapat diartikan anak yang tidak dapat mendengar. Tidak dapat mendengar tersebut dapat dimungkinkan kurang dengar atau tidak mendengar sama sekali. Secara fisik, anak tunarungu tidak berbeda dengan anak dengar pada umumnya, sebab orang akan mengetahui bahwa anak menyandang ketunarunguan pada saat berbicara, anak tersebut berbicara tanpa suara atau dengan suara yang kurang atau tidak jelas artikulasinya, atau bahkan tidak berbicara sama sekali, anak tersebut hanya berisyarat. Agar dapat diperoleh pengertian yang lebih jelas tentang anak tunarungu, berikut ini dikemukakan definisi anak tunarungu oleh beberapa ahli.

Iwin Suwarman (Edja Sadjajaah. 2005: 75), pakar bidang medik, memiliki pandangan yang sama bahwa anak tunarungu dikategorikan menjadi dua kelompok. Pertama Hard of hearing adalah seseorang yang masih memiliki sisa pendengaran sedemikian rupa sehingga masih cukup untuk digunakan sebagai alat penangkap proses mendengar sebagai bekal primer penguasaan kemahiran bahasa

dan komunikasi dengan yang lain baik dengan maupun tanpa menggunakan alat bantu dengar. Kedua *The Deaf* adalah seseorang yang tidak memiliki indera dengar sedemikian rendah sehingga tidak mampu berfungsi sebagai alat penguasaan bahasa dan komunikasi, baik dengan ataupun tanpa menggunakan alat bantu dengar. Kemampuan anak tunarungu yang tergolong kurang dengar akan lebih mudah mendapat informasi sehingga kemampuan bahasanya akan lebih baik. Anak tuli yang sudah tidak mempunyai sisa pendengaran otomatis untuk mendapat informasi sulit sehingga kemampuan bahasanya kurang baik.

Sedangkan menurut Murni Winarsih (2007: 23), menyatakan tunarungu adalah seseorang yang mengalami kekurangan atau kehilangan kemampuan mendengar baik sebagian atau seluruhnya yang diakibatkan oleh tidak fungsinya sebagian atau seluruh alat pendengaran, sehingga anak tersebut tidak dapat menggunakan alat pendengarannya dalam kehidupan sehari-hari. Hal tersebut berdampak terhadap kehidupannya secara kompleks terutama pada kemampuan berbahasa sebagai alat komunikasi yang sangat penting. Gangguan mendengar yang dialami anak tunarungu menyebabkan terhambatnya perkembangan bahasa anak, karena perkembangan tersebut, sangat penting untuk berkomunikasi dengan orang lain. Berkomunikasi dengan orang lain membutuhkan bahasa dengan artikulasi atau ucapan yang jelas sehingga pesan yang akan disampaikan dapat tersampaikan dengan baik dan mempunyai satu makna, sehingga tidak ada salah tafsir makna yang dikomunikasikan.

Dari pemaparan teori di atas dapat disimpulkan bahwa anak tunarungu memiliki gangguan pendengaran yang dikelompokkan menjadi dua golongan yaitu kurang dengar dan tuli sehingga berpengaruh terhadap kemampuan bahasanya yang menyebabkan terhambatnya perkembangan Bahasa anak tersebut dalam berkomunikasi dengan orang lain dalam kehidupan sehari-hari.

Dalam penelitian ini penulis mengambil objek anak TK tunarungu di SLB-B NEGERI BANDUNG yang mana terdapat anak tunarungu Sebagian dengan artian masih dapat mendengarkan suara orang lain, dan ada juga anak tunarungu total yang sudah tidak dapat mendengar suara orang lain.

2.1.1.1 Klasifikasi Anak Tunarungu

kebanyakan orang, berarti pendengaran anak tersebut dapat dikatakan normal. Bagi tunarungu yang mengalami hambatan dalam pendengaran itu pun masih dapat dikelompokkan berdasarkan kemampuan anak yang mendengar. Lebih lanjut untuk mengetahui pengelompokkannya, penulis memaparkan sebagai berikut:

Klasifikasi anak tunarungu yang dikemukakan oleh Samuel A. Kirk (Permanarian Somad 1996: 29) adalah sebagai berikut :

- a. 0 dB : menunjukkan pendengaran optimal.
- b. 0-26 dB : menunjukkan masih mempunyai pendengarannormal.
- c. 27-40 dB : menunjukkan kesulitan mendengar bunyi-bunyi yang jauh,membutuhkan tempat duduk yang strategis letaknya dan memerlukan terapi wicara (tergolong tunarungu ringan).
- d. 41-55 dB : mengerti bahasa percakapan, tidak dapat mengikuti diskusi kelas, membutuhkan alat bantu dengar dan terapi bicara (tergolong tunarungu sedang).
- e. 56-70 dB : hanya bisa mendengar suara dari arak yang dekat, masih mempunyai sisa pendengaran untuk belajar Bahasa ekspresif ataupun reseptif dan bicara dengan menggunakan alat bantu dengar serta dengan cara yang khusus (tergolong tunarungu agak berat).
- f. 71-90 dB : hanya bisa mendengar bunyi yang sangat dekat, kadang dianggap tuli, membutuhkan pendidikan Luar biasa yang intensif, membutuhkan alat bantu mendengar (ABM) dan latihan bicara secara khusus (tergolong tunarungu berat).
- g. 91 dB keatas : mungkin sadar akan adanya bunyi atau suara dan getaran, banyak tergantung pada penglihatan daripada

pendengarannya untuk proses menerima informasi dan yang bersangkutan dianggap tuli (tergolong tunarungu berat sekali).

Dan sebagai penjabar dari teori di atas berdasarkan tingkat kehilangan ketajaman pendengaran yang diukur dengan satuan desibel (dB), klasifikasi anak tunarungu menurut Heri Purwanto (1998: 7) adalah sebagai berikut :

a. Sangat ringan (light)	25 dB - 40 dB
b. Ringan (mild)	41 dB - 55 dB
c. Sedang (moderate)	56 dB - 70 dB
d. Berat (severe)	71 dB - 90 dB
e. Sangat berat (profound)	91 dB – lebih

Dari klasifikasi di atas dapat disimpulkan bahwa anak tunarungu memiliki tingkat kemampuan mendengar yang berbeda-beda. Yang artinya Semakin tinggi kehilangan pendengarannya, maka semakin lemah kemampuan artikulasinya.

2.1.2 Pendekatan Inquiry dan Discovery

Inquiry yang dalam bahasa Inggris *inquiry discovery learning*, berarti pertanyaan, atau pemeriksaan, penyelidikan. *Inquiry* sebagai suatu proses umum yang dilakukan manusia untuk mencari atau memahami informasi.”

Ide pokok pendekatan pembelajaran inkuiri berasal dari pemikiran John Dewey (Nasution, 1992: 117). Di dalam bukunya *How We Think*, ia memperkenalkan istilah berpikir reflektif. Maksud berpikir reflektif adalah usaha yang aktif, hati-hati, dan pengujian secara tepat terhadap keyakinan seseorang atau pengetahuan tertentu berdasarkan dukungan kenyataan. Ide inilah yang kemudian dikembangkan oleh banyak pakar pendidikan dan psikologi. Berbagai istilah kemudian mereka gunakan untuk maksud yang kira-kira sama, yaitu pendekatan inkuiri.

Beberapa definisi antara lain Good (Nasution, 1992: 117), mendefinisikan inkuiri sebagai pendekatan pemecahan masalah (problem solving) dalam belajar, setiap fenomena baru yang menantang menimbulkan reaksi untuk berpikir. Goldmark (Nasution, 1992:117) mendefinisikan inkuiri sebagai pola bereaksi dalam bentuk “bertanya” yang terarah menguji suatu nilai. Menurutnya, bertanya itu sangat penting sebagai bentuk dan sebagai tanda adanya peserta didik yang aktif.

Berlandaskan Hosnan (2014:282), discovery learning adalah model pengembangan kemampuan belajar aktif pada siswa agar bisa investigasi dan mendapatkan ilmu secara mandiri. Dengan belajar aktif ini siswa juga bisa dilatih berpikir secara analisis dan problem solving sehingga ilmu pengetahuan bisa bertahan lama dalam diri siswa.

Pada sistem demikian, biasanya terdapat dua pendekatan yang dapat dilaksanakan, yaitu pendekatan yang berorientasi pada guru dan pendekatan yang berorientasi pada siswa.

Inquiry discovery learning adalah belajar mencari dan menemukan sendiri. Sistem belajar mengajar ini guru menyajikan bahan pelajaran tidak dalam bentuk yang final, tetapi anak didik diberi peluang untuk mencari dan menemukannya sendiri dengan mempergunakan teknik pendekatan pemecahan masalah. Bentuk pengajaran terutama memberi motivasi pada siswa untuk menyelidiki masalah-masalah yang ada dengan menggunakan cara-cara dan keterampilan ilmiah dalam rangka mencari penjelasan.

Johnson dalam supriyono (2009:68) membedakan discovery learning dengan inquiry learning. Discovery terdapat pengalaman yang disebut ahaa experience yang dapat diartikan nah ini dia. Inquiry learning tidak selalu sampai pada proses ini. Hal ini karena karena proses akhir discovery learning adalah penemuan, sedangkan inquiry learning proses akhirnya terletak pada kepuasan kegiatan meneliti. Discovery learning menekankan pada pengalaman seperti yang dialami oleh peneliti ketika melakukan penemuan suatu temuan. Inquiry berarti guru harus menyediakan situasi sedemikian rupa sehingga siswa didorong untuk melakukan prosedur yang digunakan oleh penelitian (rustaman, 2002:113).

Persamaan discovery learning dan inquiry learning yaitu kedua pembelajaran tersebut menekankan pada masalah kontekstual dan aktivitas penyelidikan.

Berdasarkan teori di atas dapat disimpulkan bahwa metode pendekatan inquiry dan discovery learning dapat memacu keinginan siswa untuk mengetahui, memotivasi mereka agar melanjutkan pekerjaannya hingga menemukan jawaban untuk bisa berpikir kritis dan analitis terhadap masalah di sekitarnya hingga dapat memecahkan masalah tersebut.

2.1.3 Seni Origami

Seni Origami adalah sebuah karya seni yang berasal dari Negara Jepang. Bahan untuk membuat origami adalah kertas atau kain yang berbentuk persegi. Dr Sumanto, mengatakan bahwa melipat kertas atau origami adalah salah satu teknik karya seni atau kerajinan tangan yang pada umumnya dibuat menggunakan kertas atau kain yang berbentuk persegi, dan origami ini bertujuan untuk menghasilkan berbagai bentuk mainan, hiasan, dan kreasi lainnya. Origami adalah salah satu seni yang 7 berasal dari Jepang yang menggunakan teknik melipat kertas yang dapat menampilkan bentuk dari burung, kipas, bunga dan bentuk lainnya (Honda, 1965)

2.1.3.1 Jenis jenis Origami

1. Seni Melipat Kertas Bergerak / Origami bergerak (Action Origami)
Action Origami atau yang bisa disebut seni melipat kertas bergerak adalah dimana kita membentuk objek yang bisa digerakan seperti pesawat terbang, baling baling, dan lain-lain.



Gambar 2. 1 Pesawat Origami

2. Moduler (Origami Moduler)

Moduler Origami adalah salah satu jeni teknik origami, seni ini mengubah bentuk kertas menjadi 3 dimensi. Dalam origami moduler tersusun dari beberapa 8 lipatan kertas yang membentuk sesuatu dan saling mendukung satu kertas dengan kertas lainnya



Gambar 2. 2 Modular Origami Bunga

3. Origami Basah

origami basah merupakan origami yang bahan pembuatannya menggunakan kertas yang basah, dalam keadaan basah kertas dibentuk menjadi bentuk tertentu, setelah kertas selesai dibentuk lalu dibiarkan sehingga kertas menjadi kering. Origami ini mempunyai tingkatan yang sulit karena keterampilan kita diuji

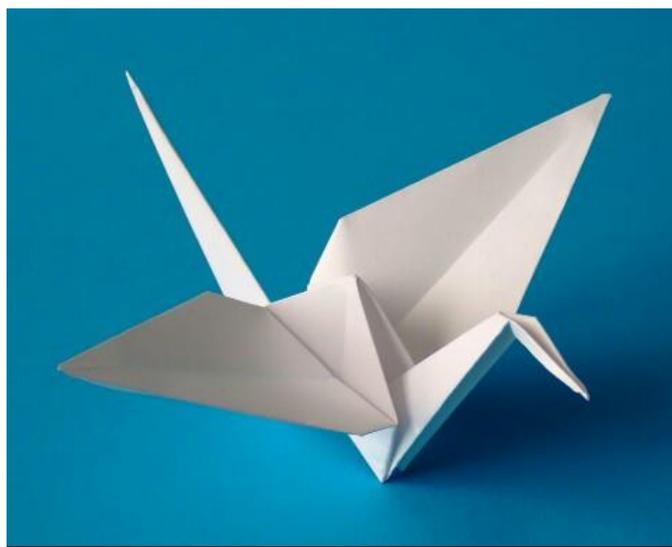
dalam membentuk suatu objek yang disebabkan bukan hanya sekedar melipat kertas membentuk objek seperti tonjolan dan lekukan yang sempurna.



Gambar 2. 3 Origami Basah

4. Origami Murni

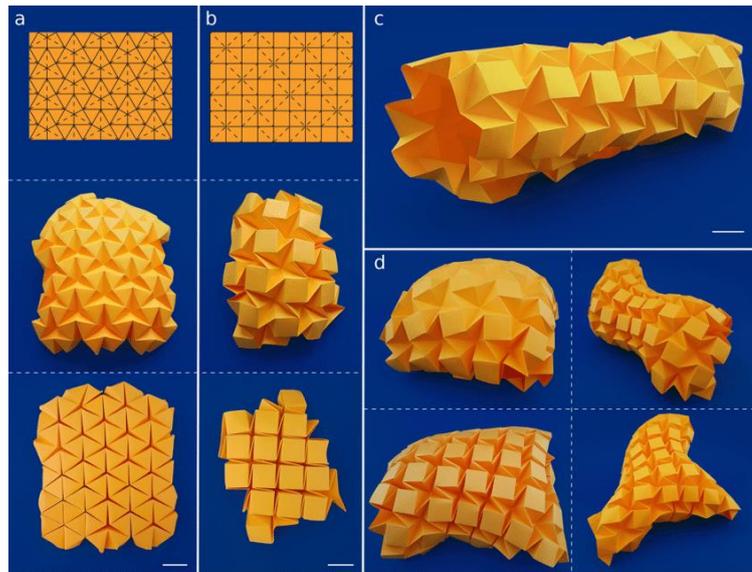
Origami murni atau Pureland origami adalah jenis origami yang sangat dasar, origami ini sudah diajarkan pada masa taman kanak-kanak (TK) dan teknik ini sangatlah umum. Teknik lipatan ini hanya menggunakan satu kertas untuk membuat sebuah model tanpa ada tambahan lem ataupun potongan



Gambar 2. 4 Origami Murni

5. Tesselasi Origami

Tesselasi adalah penyusunan beberapa bentuk origami dengan teratur tanpa adanya celah sehingga membentuk sebuah pola layaknya rajutan.



Gambar 2. 5 Origami Tesselasi

2.1.3.2 Manfaat Origami

Maya Hirai (2012) mengatakan origami tentu bukan hanya sekedar bermain melipat kertas yang mengubah selembar kertas menjadi sebuah model ada beberapa 10 manfaat yang terdapat pada origami yaitu dapat mengajarkan kreativitas, ketekunan, ketelitian serta melatih imajinasi.

2.1.4 Tumbuh Kembang Anak

Perkembangan keterampilan suatu individu, yaitu dengan cara mengembangkan system motorik, dimana anak dapat menempatkan dirinya didalam lingkungannya dan daerah sekolah. Pada usia akan masuk sekolah atau usia kelas awal Sekolah Dasar dimana itu adalah usia emas bagi anak, anak sudah dapat diajarkan menulis, membaca, menggambar.

2.1.4.1 Perkembangan Motorik

Hurlock (1978, h. 159) mengatakan bahwa perkembangan motorik

adalah kematangan untuk mengendalikan gerakan tubuh dan otak sebagai pusat gerakan. Gerakan itu dibagi menjadi dua jenis yaitu gerak kasar dan gerakan halus. Dari pendapat di atas dapat disimpulkan bahwa perkembangan motorik dapat membantu meningkatkan kinerja gerak tubuh dan otak sebagai pusat gerakan.

2.1.4.2 Pembagian Keterampilan Motorik

Richard (1989, h. 11) mengatakan bahwa keterampilan dibagi menjadi dua yaitu keterampilan motorik halus (*gross motor skill*) dan keterampilan motorik kasar (*fine motor skill*).

6. Motorik Halus (*fine motor skill*)

motorik halus (*fine motor skill*) yaitu keahlian yang menggunakan otot kecil di tubuh, motorik halus meliputi koordinasi dari mata dan tangan, keterampilan ini memerlukan kejelian yang tinggi. Contoh dari fungsi motorik halus adalah : melukis, memegang benda, melipat dan lain-lain.

7. Motorik Kasar (*gross motor skill*)

motorik kasar (*gross motor skill*) yaitu keahlian yang menggunakan otot-otot besar yang berada di tubuh. Motorik kasar meliputi melempar, berlari, melompat dan lain-lain

2.1.5 Motorik Halus

Motorik halus mempunyai peranan yang sangat penting, motorik halus adalah suatu gerakan yang hanya menggunakan anggota tubuh tertentu yang hanya dilakukan oleh otot-otot halus saja. Oleh karena itu gerakan yang ada di dalam motorik halus tidak membutuhkan tenaga yang besar melainkan hanya menggunakan koordinasi yang cermat dan hanya membutuhkan ketelitian

Saputra dan Rudyanto (2005, h. 118) menjelaskan bahawasanya motorik halus ialah suatu kemampuan anak dalam beraktivitas dengan menggunakan otot halus seperti menulis, menggambar, melipat, meremas dan lain-lain. Sedangkan

menurut Richard (1989, h. 103) mengatakan keterampilan motorik halus (fine motor skill) adalah suatu keterampilan yang membutuhkan kontrol dari otot kecil yang ada ditubuh untuk mencapai tujuan dari keterampilan. Lindya (2008) juga mengatakan motorik halus adalah aspek yang berkaitan dengan kemampuan anak untuk melakukan gerakan pada bagian tubuh tertentu saja yang dilakukan oleh oto kecil tetapi memerlukan koordinasi yang cermat.

2.1.5.1 Tujuan Peningkatan Motorik Halus

Saputra dan Rudyanto (2005, h. 115) mengatakan bahwa tujuan dari meningkatkan motorik halus adalah :

- a. Dapat memfungsikan otot kecil seperti menggerakkan tangan terutama pada jari-jari tangan.
- b. Mampu menyelaraskan kecepatan tangan dan mata.
- c. Dapat mengedalikan emosi.

Hal yang samapun dijelaskan oleh Sumantri (2005, h. 145) yang mengatakan bahwa aktivitas motorik halus pada anak dapat berkembang melalui kegiatan bermain, melipat, mewarnai, menjiplak bentuk dan lain-lain.

Dari pendapat di atas dapat disimpulkan bahwa tujuan meningkatkan motorik halus adalah mengembangkan kemampuan anak yang bertujuan mengembangkan motorik halus pada tangan dan mata yang bertujuan membantu anak tuna rungu dalam pembelajaran selanjutnya, dengan mempersiapkan kemampuannya dalam motorik halus misal menggunting, menulis, mewarnai, menempel dsb.

2.1.6 OOP

Object Oriented Programing (OOP) merupakan paradigma pemrograman yang populer saat ini yang telah menggantikan teknik pemrograman berbasis prosedur. Object Oriented Programing yang berarti pula Pemrograman Beorientasi Objek sudah ditemukan sekitar tahun 1960 dan dikembangkan pada permulaan tahun 1970. [2]

Pemrograman Berorientasi Objek (Object Oriented Programming/OOP)

merupakan pemrograman yang berorientasikan kepada objek, dimana semua data dan fungsi dibungkus dalam class-class atau object-object. Setiap object dapat menerima pesan, memproses data, mengirim, menyimpan dan memanipulasi data. Beberapa object berinteraksi dengan saling memberikan informasi satu terhadap yang lainnya.

Pemrograman adalah mengharuskan pemahaman. Bahasa pemrograman merupakan akas untuk perancangandan pemrograman. Hanya bila pemrograman dapat memahami gagasan-gagasan keberadaan fasilitas-fasilitas Dwibahasa pemrograman memungkinkan pemrogram menjadi mahir menggunakan bahasa pemrograman. Bahasa berorientasi objek adalah akas yang dapat dipergunakanoleh pemrogram secara bagus atau jelek.

Masing-masing object harus berisikan informasi mengenai dirinya sendiri dan dapat dihubungkan dengan Object yang lain. Pemrograman berorientasi objek berbeda dengan pemrograman prosedural yang hanya menggunakan satu halaman dibawah untuk mengerjakan banyak perintah atau statement. Penggunaan pemrograman berorientasi objek sangat banyak sekali, contoh : java, php, perl, c#, cobol, dan lainnya.

Konsep Dasar Pemrograman Beorientasi Objek

Berikut ini adalah konsep-konsep dalam pemrograman berorientasi objek :

1. *Class*

Kelas (*Class*) merupakan penggambaran satu set objek yang memiliki atribut yang sama. Kelas mirip dengan tipe data ada pemrograman non objek, akan tetapi lebih komprehensif karena terdapat struktur sekaligus karakteristiknya. Kelas baru dapat dibentuk lebih spesifik dari kelas ada umumnya.kelas merupakan jantung dalam pemrograman berorientasi objek.

2. *Object*

Objek merupakan teknik dalam menyelesaikan masalah yang kerap muncul dalam pengembangan perangkat lunak. Teknik ini merupakan teknik yang efektif dalam menemukan cara yang tepat dalam membangun

sistem dan menjadi metode yang paling banyak dipakai oleh para pengembang perangkat lunak. Orientasi objek merupakan teknik pemodelan sistem riil yang berbasis objek.

3. *Abstraction*

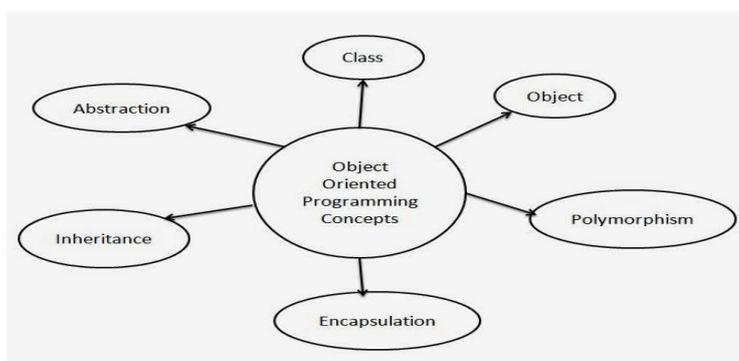
Kemampuan sebuah program untuk melewati aspek informasi yang diolah adalah kemampuan untuk fokus pada inti permasalahan. Setiap objek dalam sistem melayani berbagai model dari pelaku abstrak yang dapat melakukan kerja, laporan dan perubahan serta berkomunikasi dengan objek lain dalam sistem, tanpa harus menampakkan kelebihan diterapkan.

4. *Enkapsulasi*

Enkapsulasi adalah proses memastikan pengguna sebuah objek tidak dapat menggantikan keadaan dari sebuah objek dengan cara yang tidak sesuai prosedur. Artinya, hanya metode yang terdapat dalam objek tersebut yang diberi izin untuk mengakses keadaan yang diinginkan. Setiap objek mengakses interface yang menyebutkan bagaimana objek lainnya dapat berintegrasi dengannya. Objek lainnya tidak akan mengetahui dan tergantung kepada representasi dalam objek tersebut.

5. *Polimorfisme*

Polimorfisme merupakan suatu fungsionalitas yang diimplikasikan dengan berbagai cara yang berbeda. Pada program berorientasi objek, pembuat program dapat memiliki berbagai implementasi untuk sebagian fungsi tertentu.



Gambar 2. 6 Konsep Dasar Pemrograman Berorientasi Objek

2.1.6.1 Uml

Unified Modelling Language (UML) yaitu suatu metode permodelan secara visual untuk sarana perancangan sistem berorientasi objek, atau definisi UML yaitu sebagai suatu bahasa yang sudah menjadi standar pada visualisasi, perancangan dan juga pendokumentasian sistem *software*. Saat ini UML sudah menjadi bahasa standar dalam penulisan blue print software. [13]

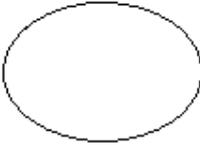
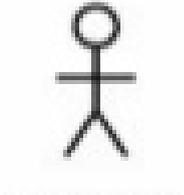
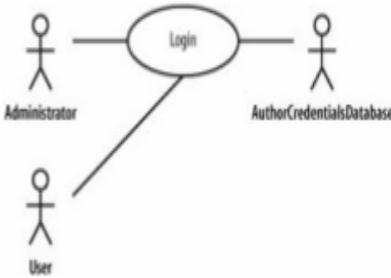
UML dibuat oleh Grady Booch, James Rumbaugh, dan Ivar Jacobson di bawah bendera *Rational Software Corps*. UML menyediakan notasi-notasi yang membantu memodelkan sistem dari berbagai perspektif. UML tidak hanya digunakan dalam pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan. Tujuan dan fungsi dari penggunaan UML di antaranya:

1. Dapat memberikan bahasa permodelan visual kepada pengguna dari berbagai macam pemrograman maupun proses rekayasa.
2. Dapat menyatukan praktik-praktik terbaik yang ada dalam permodelan.
3. Dapat memberikan model yang siap untuk digunakan, merupakan bahasa permodelan visual yang ekspresif untuk mengembangkan sistem dan untuk saling menukar model secara mudah.
4. Dapat berguna sebagai blue print, sebab sangat lengkap dan detail dalam perancangannya yang nantinya akan diketahui informasi yang detail mengenai koding suatu program.
5. Dapat memodelkan sistem yang berkonsep berorientasi objek, jadi tidak hanya digunakan untuk memodelkan perangkat lunak (*software*) saja.
6. Dapat menciptakan suatu bahasa permodelan yang nantinya dapat dipergunakan oleh manusia maupun oleh mesin

2.1.6.2 Usecase

Use case diagram yaitu salah satu jenis diagram pada UML yang menggambarkan interaksi antara sistem dan aktor, use case diagram juga dapat mendeskripsikan tipe interaksi antara pemakai sistem dengan sistemnya. [13]

Tabel 2. 1 Simbol yang digunakan dalam use case

Nama Simbol	Gambar	Keterangan
Sistem		merupakan batasan (boundary) yang isinya merupakan berbagai use case yang digunakan.
Use Case		menunjukkan kegiatan/aktivitas yg bisa dilakukan.
Aktor		Entitas luar yang bisa memicu use case yang terhubung ke sistem, atau bisa dikatakan aktor adalah pengguna yang harus berinteraksi dengan sistem yang dibuat.
Relasi		Garis penghubung antara aktor dan use case maupun use case dengan use case yang lain, tanda include diantara garis yang terdapat antar use case menunjukkan bahwa use case tersebut membutuhkan use case lain yang sudah dihubungkan kepadanya, sedangkan tanda extend merupakan opsi alternatif bila ingin digunakan.

2.1.6.3 Usecase Scenario

Setiap use case diagram dilengkapi dengan skenario, skenario use case / use case skenario adalah alur jalannya proses use case dari sisi aktor dan system. Berikut adalah format tabel skenario *use case*.

Tabel 2. 2 Keterangan dalam use case scenario

Deskripsi	Keterangan
Nama use case	menunjukkan kegiatan yang dilakukan.
Goal in context	hal yang bisa didapatkan oleh actor.
Description	penjelasan use case berdasarkan fungsionalitas.
Relasi use case	jika ada hubungan yang dibutuhkan dengan use case lain.
precondition	kondisi yang diperlukan sebelum menjalankan use case.
post-condition	kondisi yang ada setelah use case dijalankan, ada dua kondisi yaitu berhasil/sukses dan gagal.
Aktor	memicu use case untuk dijalankan.
Trigger	digunakan oleh aktor agar bisa menjalankan use case yang dipilih.
Main flow	alur interaksi antara aktor dan sistem setelah menjalankan use case yang menunjukkan jika kondisinya benar.
extension	kebalikan dari main flow, yaitu alur yang menunjukkan jika kondisinya salah.

2.1.6.4 Activity Diagram

Activity diagram atau diagram aktivitas yaitu salah satu jenis diagram pada UML yang dapat memodelkan proses-proses apa saja yang terjadi pada sistem. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. [2] Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut :

1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan
2. Urutan atau pengelompokan tampilan dari sistem/user interface dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.

Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya

SIMBOL	SEMANTIC	FUNGSI
	Initial State	Penanda awal dari aktivitas
	Final State	Penanda akhir dari aktivitas
	Transisi	Menggambarkan alur antar aksi
	Aksi	Menggambarkan aksi yang ada pada satu aktivitas

Gambar 2. 7 Simbol Activity Diagram

2.1.6.5 Class Diagram

Class diagram yaitu salah satu jenis diagram pada UML yang digunakan untuk menampilkan kelas-kelas maupun paket-paket yang ada pada suatu sistem yang nantinya akan digunakan. Jadi diagram ini dapat memberikan sebuah gambaran mengenai sistem maupun relasi-relasi yang terdapat pada sistem tersebut. [13] Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut :

1. Kelas Main yaitu kelas yang memiliki fungsi awal di eksekusi ketika sistem dijalankan

2. Kelas Interface kelas yang mendefinisikan dan mengatur tampilan ke pemakai, biasanya disebut kelas *boundaries*.
3. Kelas yang di ambil dari pendefinisian *use case* merupakan kelas yang menangani fungsi-fungsi yang harus diambil dari pendefinisian *use case*
4. Kelas Entitas merupakan kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data.

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
2		<i>N-Ary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

Gambar 2. 8 Simbol Class Diagram

2.1.4.1 Class Diagram

Sequence diagram yaitu salah satu jenis diagram pada UML yang menjelaskan interaksi objek yang berdasarkan urutan waktu, *sequence* diagram

juga dapat menggambarkan urutan atau tahapan yang harus dilakukan untuk dapat menghasilkan sesuatu seperti pada *use case diagram*. [13]

Nama	Simbol	Keterangan
Actor		Menggambarkan seseorang atau sesuatu yang berinteraksi dengan sistem.
Boundary		Menggambarkan interaksi antara satu atau lebih yang menjadi penghubung antara actor dengan sistem.
Control		Menggambarkan “perilaku mengatur”, mengkoordinasikan perilaku sistem dan dinamika dari suatu sistem, menangani tugas utama dan mengontrol alur kerja suatu sistem.
Entity		Menggambarkan informasi yang harus disimpan oleh sistem, entity juga memperlihatkan struktur data dari sebuah sistem.
Activation		Menggambarkan eksekusi terhadap obyek. Panjang kotak ini berbanding lurus dengan durasi aktivasi sebuah operasi.
Message		Menggambarkan message antar obyek, yang menunjukkan urutan kejadian yang terjadi.
Message to Self		Looping terhadap obyek itu sendiri. Menggambarkan pesan atau hubungan obyek itu sendiri, yang menunjukkan urutan kejadian yang terjadi.
Loop		Menunjukkan perilaku perulangan atau menggambarkan dari suatu kegiatan yang berulang-ulang

Tabel 2. 3 Simbol Sequence Diagram

2.1.7 Aplikasi

Aplikasi adalah program yang dibuat oleh pemakai yang ditujukan untuk melakukan suatu tugas khusus (Kadir, 2003).

Menurut Kadir (2008:3) program aplikasi adalah program siap pakai atau

program yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain. Aplikasi juga diartikan sebagai penggunaan atau penerapan suatu konsep yang menjadi pokok pembahasan atau sebagai program komputer yang dibuat untuk menolong manusia dalam melaksanakan tugas tertentu. Aplikasi software yang dirancang untuk penggunaan praktisi khusus, klasifikasi luas ini dapat dibagi menjadi 2 (dua) yaitu:

- a. Aplikasi software spesialis, program dengan dokumentasi tergabung yang dirancang untuk menjalankan tugas tertentu.
- b. Aplikasi paket, suatu program dengan dokumentasi tergabung yang dirancang untuk jenis masalah tertentu.

Dari kedua pengertian pendapat di atas dapat disimpulkan bahwa aplikasi adalah sekumpulan perintah atau kode yang disusun secara sistematis untuk menjalankan suatu perintah yang diberikan oleh manusia melalui komponen atau hardware komputer yang digunakan oleh manusia dalam menjalankan program aplikasi, dengan demikian bisa membantu manusia untuk memberikan solusi dari apa yang diinginkan.

2.1.8 Mobile Development

Android Studio adalah sebuah Integrated Development Environment (IDE) untuk Android Development yang diperkenalkan google pada acara Google I/O 2013. Android Studio merupakan pengembangan dari Eclipse IDE, dan dibuat berdasarkan IDE Java populer, yaitu IntelliJ IDEA. Android Studio merupakan IDE resmi untuk pengembangan aplikasi Android. [13]



Gambar 2. 9 Logo Android Studio

Sebagai pengembangan dari Eclipse, Android Studio mempunyai banyak fitur-fitur baru dibandingkan dengan Eclipse IDE. Berbeda dengan Eclipse yang menggunakan Ant, Android Studio menggunakan Gradle sebagai build environment. Fitur-fitur lainnya adalah sebagai berikut :

1. Sistem Pembuatan berbasis Gradle yang fleksibel
2. Emulator yang cepat dan kaya fitur
3. Lingkungan yang menyatu untuk pengembangan bagi semua perangkat android
4. Instant Run untuk mendorong perubahan ke aplikasi yang berjalan tanpa membuat APK baru
5. Template kode dan integrasi GitHub untuk membuat fitur aplikasi yang sama dan mengimpor kode contoh
6. Alat penguji dan kerangka kerja yang ekstensif
7. Alat Lint untuk meningkatkan kinerja, kegunaan, kompatibilitas versi dan masalah-masalah lain
8. Dukungan C++ dan NDK
9. Dukungan bawaan untuk Google Cloud Platform, mempermudah pengintegrasian Google Cloud Messaging dan App Engine

2.1.9 Firebase

Firebase adalah BaaS (*Backend as a Service*) yang saat ini dimiliki oleh Google. Firebase ini merupakan solusi yang ditawarkan oleh Google untuk mempermudah pekerjaan *Mobile Apps Developer*. Dengan adanya Firebase, apps developer bisa fokus mengembangkan aplikasi tanpa harus memberikan effort yang besar untuk urusan backend [17].

Beberapa fitur yang dimiliki oleh Firebase adalah sebagai berikut :

1. Firebase Analytics.
2. Firebase Cloud Messaging dan Notifications.
3. Firebase Authentication.
4. Firebase Remote Config.
5. Firebase Real Time Database.

6. Firebase Crash Reporting.

Dua fitur yang menarik adalah Firebase Remote Config dan Firebase Real Time Database. Secara sederhananya, Remote Config adalah fitur yang memungkinkan developer mengganti / mengubah beberapa konfigurasi aplikasi Android / iOS tanpa harus memberikan update aplikasi via Play Store / App Store. Salah satu konfigurasi yang bisa dimanipulasi adalah seperti warna / tema aplikasi.

Sedangkan Firebase *Real Time Database* adalah fitur yang memberikan sebuah NoSQL database yang bisa diakses secara *Real Time* oleh pengguna aplikasi. Dan hebatnya adalah aplikasi bisa menyimpan data secara lokal ketika tidak ada akses internet, kemudian melakukan sync data segera setelah mendapatkan akses internet.