BAB 2
LANDASAN TEORI

2.1 Eagle School

Eagle School atau sekolah Eagle adalah sebuah lembaga pendidikan swasta yang berlokasi di Lembang Bandung, sekolah ini memiliki visi dan misi sebagai berikut :

2.1.1 Visi

Berikut ini merupakan visi dari sekolah Eagle yang terpapar pada situs sekolah mereka :

1. *Excellent Achievement*
   To be what God wants us to be. To do what God wants us to do. To have what God wants us to have
2. *Global*
   God's Standard. Global Impact
3. *Leadership*
   Creative Man. Cultural Mandate
4. *Education*
   Discipleship (Christian World View)

2.1.2 Misi

Berikut merupakan misi yang dimiliki oleh sekolah Eagle :

1. *School of Lord (Christian World View)*
   Training disciples with Godly character-based curriculum.

2. *School of Life (Christian World Laboratory)*
   Encouraging children to maximize their potential through excellent basic and life skill learning experiences.

3. *School of Love (Christian World Atmosphere)*
   Showing example and educating children to love God and serving others passionately.

2.2 Aplikasi

Aplikasi adalah sebuah program komputer yang dibuat untuk mengerjakan suatu pekerjaan dalam rangka membantu pekerjaan manusia dimana harapanya dengan bantuan aplikasi tersebut pekerjaan dapat lebih cepat, akurat, efisien dan lain sebagainya. Sekarang ini penerapan aplikasi tidak hanya pada dunia industri atau perkantoran saja namun sudah masuk ke berbagai hal seperti pendidikan, hiburan dan masih-banyak lagi, salah satu contoh aplikasi yang sangat popular adalah Microsoft Office, sebenarnya Microsoft office adalah sebuah suite aplikasi (*Application suite*) dimana di dalamnya terdapat banyak aplikasi dengan peruntukan yang berbeda-beda seperti Power Point, Word, Exel dan lain-lain, dengan Power Point contohnya, pengguna dapat membuat sebuah tayangan presentasi yang sangat bervariatif dan tidak terbatas perancangannya, di dalam aplikasi tersebut juga disematkan fitur untuk menampilkan video, musik, kurva dan masih banyak lagi fitur yang disediakan untuk membantu penampilan tayangan slide, itu merupakan contoh kecil dari sebuah aplikasi yang sudah digunakan banyak orang dan bertujuan untuk membantu pekerjaan manusia.

2.3 Android

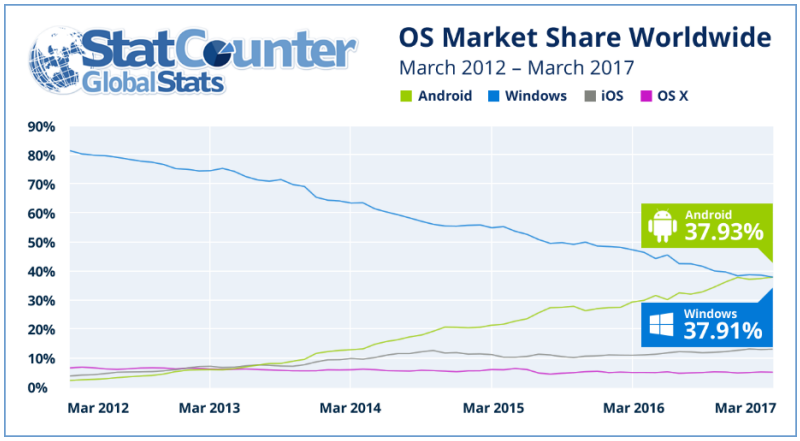


Gambar 2.1 Logo Android

Android adalah sebuah perangkat lunak untuk ponsel, Tabelt dan berbagai banyak perangkat lainnya yang terus dikembangkan penerapannya, android

merupakan proyek *open source* (di pimpin oleh Google namun bukan kepunyaan mereka) disebut AOSP (*Android Open Source Project*). Google menggunakan projek ini untuk membuat dasar atau basis agar dapat membuat versi Android milik mereka sendiri yang kemudian digunakan oleh parbrikan lain.

Android merupakan sebuah OS *open source* yang sangat menguntungkan bagi para developer untuk dapat menciptakan inovasi-inovasi dalam membuat sebuah aplikasi, dengan pengguna yang sangat menjanjikan kini android menjadi salah satu OS yang paling banyak sigunakan di dunia, android juga sangat banyak digunakan oleh berbagai pabrikan ponsel untuk dijadikan OS(*operating system*) ponsel-ponsel mereka, tak heran jika perkembangan pengguna android saat ini sangat meningkat pesat, seperti ditunjukan dalam **Gambar 2.2 Statistik pengguna OS** berikut yang dilansir dari gs.statcounter.com



Gambar 2.2 Statistik pengguna OS

2.4 XML

XML (*Extensible Markup Language*) adalah bahasa markup untuk keperluan umum yang disarankan oleh W3C untuk membuat dokumen markup keperluan pertukaran data antar sistem yang beraneka ragam.XML merupakan kelanjutan dari HTML (*HyperText Markup Language*) yang merupakan bahasa standar untuk melacak Internet. Teknologi XML adalah teknologi keturunan dari SGML (*Standard Generalized Markup Language*, ISO 8879) yang dikembangkan pada tahun 1980-an. XML adalah suatu bahasa Markup, yaitu bahasa yang berisikan kode-kode berupa tanda-tanda tertentu dengan aturan tertentu untuk memformat

dokumen teks dengan tag sendiri agar dapat dimengerti. XML adalah language untuk mengidentifikasi dan menandai data terstruktur

XML didesain untuk mempu menyimpan data secara ringkas dan mudah diatur.Kata kunci utama XML adalah data (jamak dari datum) yang jika diolah bisa memberikan informasi. XML menyediakan suatu cara terstandarisasi namun 27 bisa dimodifikasi untuk menggambarkan isi dari dokumen. Dengan sendirinya, XML dapat digunakan untuk menggambarkan sembarang view database, tetapi dengan suatu cara yang standar.

XML memiliki tiga tipe file, diantaranya:

1. XML, merupakan standar format dari struktur berkas (file).
2. XSL, merupakan standar untuk memodifikasi data yang diimpor atau diekspor.
3. XSD, merupakan standar yang mendefinisikan struktur database dalam XML.

Keunggulan XML bisa diringkas sebagai berikut :

1. Pintar (Intelligence). XML dapat menangani berbagai tingkat (level) kompleksitas.
2. Dapat beradaptasi. Dapat mengadaptasi untuk membuat bahasa sendiri. Seperti Microsoft membuaat bahasa MSXML atau Macromedia mengembangkan MXML.
3. Mudah pemeliharaannya.
4. Sederhana. XML lebih sederhana.
5. Mudah dipindah-pindahkan (Portability). XML mempunyai kemudahan perpindahan (portabilitas) yang lebih bagus.

2.5 *Multiple intelligences*

Howard Gardner merupakan seorang tokoh dalam dunia psikologi yang terkenal karena pandangan atau teori kecerdasan yang di ungkapkannya dalam bukunya yang berjudul "*Frames of mind: The Theory of Multiple intelligences*", menurut Howard Gardner, kecerdasan seseorang meliputi unsur - unsur kecerdasan *Logical-Mathematical, Musical, Naturalist, Spatial, Intra-Personal, Linguistic,*

Bodily-Kinesthetic,dan *Inter-Personal* [5], dalam buku Howard Gardner [3] yang berjudul "*Multiple intelligences*" berikut bagaimana Gardner menerangkat setiap kecerdasan yang berhasil ia dapatkan :

1. *Logical-Mathematical Intelligence*

Individu yang berbakat (memiliki kecerdasan *Logical-Mathematical*), proses pemecahan sebuah masalah itu terjadi luar biasa cepat, ilmuan yang sukses memecahkan banyak variabel sekaligus dan menciptakan banyak hipotesis yang masing-masing dievaluasi lalu diterima atau ditolak secara bergantian. Bentuk kecerdasan ini telah diinvestigasi dengan seksama oleh para psikolog tradisional dan merupakan ciri utama bagi "kecerdasan mentah" atau kemampuan pemecahan masalah yang sepertinya ditemukan di berbagai domain, selain itu wilayah-wilayah tertentu pada otak lebih mengutamakan kalkulasi matematis dibandingkan subjek lainnya.

2. Musical Intelligence

Ketika Yehudi Menuhin berusia tiga tahun, orangtuanya membawanya ke konser *San Francisco Orchestra*. Suara biola Louis Persinger begitu memesona anak kecil itu sehingga ia bersikukuh untuk mendapatkan biola di hari ulang tahunnya dan meminta Louis Persinger menjadi gurunya. Menuhin mendapatkan keduanya. Saat ia berusia sepuluh tahun, menuhin menjadi pemain biola internasional.

Kecerdasan musikal pemain biola Yehudi Menuhin termanifestasi sendiri bahkan sebelum ia menyentuh biola atau menerima pelatihan musik apa pun. Reaksi kuatnya pada suara tertentu dan kemajuan pesatnya pada instrumen itu menunjukan bahwa ia secara biologis dipersiapkan dalam cara tertentu untuk hidup dalam dunia musik.

3. Naturalist Inteligence

Orang yang memiliki tingkat kecerdasan naturalis yang tinggi amat sadar akan bagaimana membedakan tanaman, hewan, pegunungan,atau konfigurasi awan yang berbeda dalam ceruk ekologis mereka. Kapasitas kecerdasan ini

tidak selalu berkaitan dengan fakor visual; mengenali nyanyian burung tertentu atau teriakan ikan paus memerlukan persepsi pendengaran.

4. Spatial Inteligence

Navigasi di sekitar Kepulauan *Caroline* di Lautan Selatan dilakukan oleh para pelayan pribumi tanpa instrumen apa pun. Posisi bintang, yang dilihat dari berbagai pulau, pola cuaca dan warna air adalah penanda utama. Tiap perjalanan dibagi ke dalam serangkaian segmen, dan navigator mempelajari posisi bintang disetiap masing-masing segmen ini. Selama perjalanan aktual navigator harus membayangkan sebuah pulau sebagai referensi sewaktu melewati bintang tertentu. Saat melakukan khayalan itu, ia menghitung jumlah segmen yang terselesaikan, proporsi perjalanan yang tersisa dan segala korelasi dalam arah yang diperlukan. Navigator tidak bisa melihat kepulauan ketika ia berlayar; sebaliknya ia memetakan lokasi kepulauan tersebut dalam gambaran mental perjalanan itu, seni visual juga menggunakan kecerdasan ini dalam penggunaan ruang. Populasi tunanetra memberikan ilustrasi perbedaan antara kecerdasan spasial dengan persepsi visual. Seorang yang buta bisa mengenali bentuk dengan metode non-visual: meraba sepanjang kontur suatu benda diterjemahkan ke dalam panjangnya waktu gerakan, yang selanjutnya diterjemahkan kedalam bentuk dan ukuarn benda itu.

5. Intra-Personal Inteligence

Pengetahuan aspek-aspek internal dari seseorang: akses pada kehidupan perasaan seseorang sendiri, rentang emosi seseorang, kapasitas untuk membedakan emosi-emosi ini dan akhirnya memberi label dan menggunakanya sebagai saran pemahaman dan memandu perilaku orang itu sendiri. Seseorang yang memliki kecerdasan intrapersonal memiliki model efektif dan layak dari dirinya sendiri yang konsisten dengan deskripsi yang dibentuk oleh pengamat yang seksama yang mengenal orang itu secara dekat.

6. Linguistic intelligence

Seperti halnya kecerdasan logis, menyebut keahlian linguistik sebagai kecerdasan juga konsisten dengan pandangan psikologi tradisional. Kecerdasan linguistik juga lulus dalam tes empiris kami. Misalnya, wilayah spesifik otak, yang disebut juga *Broca,* bertanggung jawab atas produksi kalimat gramatikal. Seseorang yang mengalami kerusakan pada wilayah ini bisa memahami kata dan kalimat dengan cukup baik tapi mengalami kesulitan dalam menyatukan kata-kata menjadi kalimat apapun selain bentuk sederhana sebuah kalimat.

7. Bodily-Kinesthetic Intelligence

Evolusi pergerakan tubuh yang khusus adalah keuntungan nyata pada spesies yang bersangkutan, dan pada manusia, adaptasi ini diperluas melalui penggunaan alat. Pergerakan tubuh menjalani jadwal perkembangan tertentu yang jelas pada anak-anak; ada sedikit pertanyaan tentang keuniversalannya pada semua budaya. Karena itu, keahliannya "pengetahuan" kinestetik tubuh memenuhi banyak kriteria sebagai suatu bentuk kecerdasan

8. Inter-Personal Intelligence

Kecerdasan interpersonal berkembang pada kapasitas inti untuk memerhatikan perbedaan di antara orang lain, perbedaan suasana hati, tempramen, memotivasi dan niat mereka. Dalam bentyj yang lebih lanjut, kecerdsan ini memungkinkan orang dewasa yang terlatih untuk membaca niat dan hasrat orang lain, bahkan ketika semua itu tersebunyi. Keahlian ini muncul dalam bentuk yang amat kompleks dalam pemimpin politik atau agama, wiraniaga, pemasar, guru, ahli terapi dan orangtua.

2.6 Multiple Intelligence Di Dunia Pendidikan

Ketika Howard Gardner, Ph.D., mencetuskan teori *Multiple Intelligence* dalam buku *Frames of Mind*, MI masih dalam ranah teori dunia psikologi, namun seiring berjalannya waktu, Howard mulai menerapkan sekaligus mensosialisasikan teori kecerdasan ganda ini ke beberapa sekolah dan tentunya banyak sekali tantangan yang dia hadapi dalam menerapkan teori kecerdasan ganda tersebut, di Indonesia kecerdasan ganda juga sudah mulai dilirik oleh beberapa sekolah bahkan ada yang sudah menerapkan nya, salah satu tokoh yang penulis kutip adalah Munif Chatif asal indonesia, seseorang yang sangat bisa dikatakan cukup kompeten ketika membahas tentang dunia pendidikan tidak terkecuali metode pembelajaran berbasis multiple intelligence, dalam bukunya yang berjudul "Sekolahnya Manusia", beliau menyampaikan bahwa menerapkan MI di indonesia akan cukup banyak tantangan dan hambatan namun Munif Chatif selalu mendapatkan dukungan langsung dari pencetus teori MI yaitu Howard Gardner sendiri.

2.7 Multiple Intelligence sebagai Strategi Pembelajaran

Multiple Intelligence memang masih bisa dikatakan sesuatu hal yang baru dalam dunia pendidikan, termasuk di Indonesia ada banyak sekali sekolah yang masih kebingungan dengan Kecerdasan Ganda ini, Munif Chatif juga menyampaikan dalam bukunya bahwa hasil penelitian yang Ia lakukan pada 2003 terhadap sekolah-sekolah di Indonesia yang menerapkan *Multiple intelligences* memberikan hasil bahwa hampir semua sekolah tersebut terjebak pada pemahaman bahwa MI adalah suatu bidang studi, seperti kecerdasan linguistic dianggap sebagai bidang studi bahasa indonesia, kecerdasan musik dianggap sebagai bidang studi musik, bernyanyi dan lain-lain, sebenarnya multiple intelligence adalah sebuah strategi pembelajaran yang digunakan untuk materi apapun dan tidak terikat oleh kecerdasan pada MI, lantas apa yang dimaksud dengan strategi pembelajaran berbasis MI ?, Munif Chatif menyampaikan dalam bukunya bahwa strategi pembelajaran berbasis MI pada intinya adalah bagaimana guru dapat mengemas gaya mengajarnya agar mudah dimengerti dan ditangkap oleh siswanya [6].

2.8 Menerapkan MI di Sekolah

Dalam buku "Sekolahnya Manusia" [6] penulis menyimpulkan bahwa menerapkan metode MI di sekolah terdiri dari 3(tiga) tahapan yaitu :

1. Analisis Jenis Kecerdasan

   Tahapan ini merupakan tahapan yang menjadi pusat penelitian dari tulisan akhir penulis, pada fase ini para guru harus mengetahui apakah yang menjadi kecerdasan para siswanya.

2. Model Aktivitas

   Ketika jenis kecerdasan dari tiap-tiap siswa sudah terindikasi maka berikutnya para guru dapat membuat model aktivitas atau mengajar, sesuai dengan kecerdasan masing-masing siswanya

3. Penerapan

   Langkah akhir tentu saja penerapan daripada model aktivitas yang telah dibut dan disiapkan sebelumnya

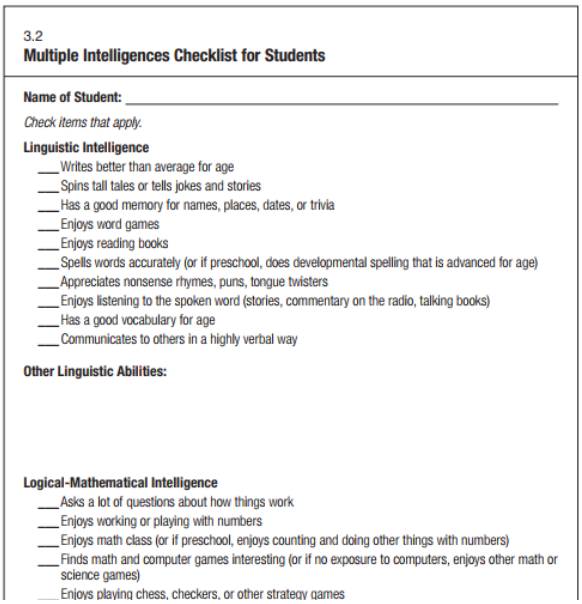
2.9 Menilai Kecerdasan Siswa

Thomas Armstrong, Ph.D., menyebutkan dalam bukunya yang berjudul "*Multiple Intelligance in the classroom"* bahwa tidak ada semacam tes yang dapat dilakukan dalam waktu 15 menit yang bisa menjelaskan kecerdasan anak, semuanya adalah tentang pengamatan terhadap sikap siswa dalam berbagai hal [7], walaupun begitu dalam buku tersebut Thomas Armstrong juga menjelaskan beberapa cara yang dapat dijadikan sebagai penilaian multiple intelligance dimana penulis akan menerapkannya dalam sekolah eagle school, berikut beberapa cara yang dikemukakan oleh Thomas Armstrong dalam bukunya terkait cara penilaian MI siswa yang akan penulis terapkan :

1. *Checklist*

   Untuk membantu guru dalam mengorganisir penelitian atau pengamatan siswa, Thomas Armstrong menyarankan metode *cheklist,* untuk membantu menentukan dan melihat perkembangan kecerdasan dari siswa tersebut , karena metode cheklist ini bersifat untuk membantu penilaian serta melihat kearah mana kecerdasan siswa tersebut maka cara mengambil kesimpulan dari metode

ini adalah dengan melihat manakah hasil ceklis yang lebih banyak, berikut Gambar 2.3 Contoh Lembar Cheklist yang ada pada buku Thomas Armstrong dan juga digunakan oleh sekolah Eagle saat ini :



3.2
Multiple Intelligences Checklist for Students

Name of Student: _____

Check items that apply.

Linguistic Intelligence

___Writes better than average for age
___Spins tall tales or tells jokes and stories
___Has a good memory for names, places, dates, or trivia
___Enjoys word games
___Enjoys reading books
___Spells words accurately (or if preschool, does developmental spelling that is advanced for age)
___Appreciates nonsense rhymes, puns, tongue twisters
___Enjoys listening to the spoken word (stories, commentary on the radio, talking books)
___Has a good vocabulary for age
___Communicates to others in a highly verbal way

Other Linguistic Abilities:

Logical-Mathematical Intelligence

___Asks a lot of questions about how things work
___Enjoys working or playing with numbers
___Enjoys math class (or if preschool, enjoys counting and doing other things with numbers)
___Finds math and computer games interesting (or if no exposure to computers, enjoys other math or science games)
___Enjoys playing chess, checkers, or other strategy games

Gambar 2.3 Contoh Lembar *Checklist*

2. Ask Student

Thomas Armstrong menyebutkan bahwa siswa merupakan ahli tak terbatas dalam kecerdasan ganda, karena mereka sendirilah yang mengenal kepribadian mereka sendiri [7], dengan metode ini kita dapat melakukan sebuah tes dimana siswa menjawab langsung setiap pertanyaan yang diajukan, tes tersebut dapat berupa *interview* atau sebuah test, penulis akan menggunakan sebuah test yang dibuat oleh Dr. Siti Sriyati, M.Pd yang diadaptasi dari "*The Rogers Indicator of Multiple intelligences (RIMI) Test"* [8].

Berikut adalah contoh pertanyaan yang dibuat oleh Dr. Siti Sriyati, M.Pd

Tabel 2.1 Soal MI

No.	Pertanyaan	Skor				
		Jarang	**Kadang**	**Cukup Sering**	**Sering**	**Selalu**
1.	Saya berhati-hati dalam mengungkapkan makna tersirat dan tersurat dari kata-kata yang saya gunakan dalam percakapan dan tulisan.	1	2	3	4	5
2.	Saya menyukai berbagai macam jenis musik.	1	2	3	4	5

Siswa harus mengisi setiap pertanyaan pada skor yang paling sesuai dengan dirinya, setelah semua pertanyaan terjawab maka saatnya melakukan proses perhitungan dengan menghitung setiap nilai pada masing-masing kecerdasan

Tabel 2.2 Skor Tes MI

Jenis Kecerdasan	No.	Skor	No.	Skor	No.	Skor	No.	Skor	No.	Skor	No.	Skor	No.	Skor	TOTAL
Linguistik	1.		9.		17.		25.		33.		41.		49.		
Musikal	2.		10.		18		26.		34.		42.		50.		
Logika-Matematika	3.		11.		19.		27.		35.		43.		51.		
Spasial	4.		12.		20.		28.		36.		44.		52.		
Kinestetik	5.		13.		21.		29.		37.		45.		53.		
Intrapersonal	6.		14.		22.		30.		38.		46.		54.		
Interpersonal	7.		15.		23.		31.		39.		47.		55.		
Naturalistik	8.		16.		24.		32.		40.		48.		56.		

Untuk mengambil kesimpulan dari tes ini kita tinggal melihat total poin mana yang lebih tinggi, selain itu dalam makalah Dr. Siti Sriyati, M.Pd juga menjelaskan keterangan dari masing-masing skor berdasarkan area nilai seperti berikut ini :

a) Skor ≤ 15

Kecerdasan ini bukan merupakan kecerdasan yang menonjol. Anda mungkin akan menghindari kegiatan yang berhubungan dengan penggunaan kecerdasan ini. Diperlukan usaha yang lebih untuk menjadi ahli pada kecerdasan ini.

b) 15 < Skor < 27

Anda nyaman dan mudah menggunakan kecerdasan ini. Anda dapat mengaplikasikan atau bahkan tidak menggunakannya. Ketika Anda menerimanya, Anda tidak dapat begitu menguasainya. Kegiatan yang menggunakan kecerdasan ini mungkin akan memberikan hasil yang memuaskan. Akan tetapi, anda juga masih membutuhkan banyak upaya.

c) Skor > 27

Kecerdasan tersebut merupakan kecerdasan yang paling menonjol yang dapat dengan mudah Anda gunakan. Anda lebih dihargai dan bermanfaat dengan menjadi ahli dalam kecerdasan ini. Keahlian akan membutuhkan sedikit usaha saja.

2.10 Contoh Perhitungan MI

Berikut ini merupakan sebuah contoh tentang bagaimana sebuah perhitungan MI dapat diambil untuk ditarik kesimpulan mengenai kecerdasan mana yang dimiliki siswa

1. Metode *Checklist*

   Untuk metode *Checklist* sendiri kita hanya perlu melihat jumlah ceklis dari setiap kecerdasannya.

2. Metode *Ask Student*

   Misalkan contoh hasil tes adalah sebagai berikut

Tabel 2.3 Contoh Hasil Jawaban

No Soal	Jawaban (1 - 5)	Keterangan
1	1	Jarang
2	3	Cukup sering
3	2	Kadang

4	4	Sering
5	1	Jarang
6	5	Selalu
7	2	Kadang
8	2	Kadang
9	1	Jarang
…	…	…

Langkah berikutnya kita tinggal masukan setiap jawaban tersebut ke Tabel berikut ini sesuai dengan soalnya, lalu hitung total skor masing-masing kecerdasan, total skor yang lebih tinggi menunjukan kecerdasan yang lebih dominan

Tabel 2.4 Tabel menghitung soal

Jenis Kecerdasan	No.	Skor	No.	Skor	No.	Skor	No.	Skor	No.	Skor	No.	Skor	No.	Skor	**TOTAL**
Linguistik	1.	**1**	9.	**1**	17.	**3**	25.	**1**	33.	**1**	41.	**1**	49.	**1**	9
Musikal	2.	**3**	10.	**3**	18	**1**	26.	**1**	34.	**3**	42.	**2**	50.	**2**	18
Logika-Matematika	3.	**2**	11.	**1**	19.	**5**	27.	**3**	35.	**2**	43.	**3**	51.	**4**	20
Spasial	4.	**4**	12.	**3**	20.	**5**	28.	**4**	36.	**1**	44.	**5**	52.	**5**	27
Kinestetik	5.	**1**	13.	**2**	21.	**1**	29.	**2**	37.	**5**	45.	**2**	53.	**2**	15
Intrapersonal	6.	**5**	14.	**3**	22.	**3**	30.	**2**	38.	**2**	46.	**3**	54.	**1**	19
Interpersonal	7.	**2**	15.	**4**	23.	**1**	31.	**1**	39.	**5**	47.	**1**	55.	**3**	17
Naturalistik	8.	**2**	16.	**1**	24.	**1**	32.	**1**	40.	**5**	48.	**2**	56.	**5**	17

Kesimpulan dari hasil di atas adalah siswa tersebut memiliki kecerdasan Spasial yang lebih dominan.

2.11 UML

UML (Unified Modeling Language) adalah sebuah bahasa yang berdasarkan grafik/gambar untuk memvisualisasi, menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan software berbasis OO

(Object-Oriented). UML sendiri juga memberikan standar penulisan sebuah sistem blue print, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema database, dan komponen-komponen yang diperlukan dalam sistem software. Pendekatan analisa & rancangan dengan menggunakan model OO mulai diperkenalkan sekitar pertengahan 1970 hingga akhir 1980 dikarenakan pada saat itu aplikasi software sudah meningkat dan mulai komplek. Jumlah yang menggunakaan metoda OO mulai diuji cobakandan diaplikasikan antara 1989 hingga 1994, seperti halnya oleh Grady Booch dari Rational Software Co., dikenal dengan OOSE (Object-Oriented Software Engineering), serta James Rumbaugh dari General Electric, dikenal dengan OMT (Object Modelling Technique). Kelemahan saat itu disadari oleh Booch maupun Rumbaugh adalah tidak adanya standar penggunaan model yang berbasis OO, ketika mereka bertemu ditemani rekan lainnya Ivar Jacobson dari Objectory mulai mendiskusikan untuk mengadopsi masing-masing pendekatan metoda OO untuk membuat suatu model bahasa yang uniform / seragam yang disebut UML (Unified Modeling Language) dan dapat digunakan oleh seluruh dunia. Secara resmi bahasa UML dimulai pada bulan oktober 1994, ketika Rumbaugh bergabung Booch untuk membuat sebuah project pendekatan metoda yang uniform/seragam dari masing-masing metoda mereka. Saat itu baru dikembangkan draft metoda UML version 0.8 dan diselesaikan serta di release pada bulan oktober 1995. Bersamaan dengan saat itu, Jacobson bergabung dan UML tersebut diperkaya ruang lingkupnya dengan metoda OOSE sehingga muncul release version 0.9 pada bulan Juni 1996. Hingga saat ini sejak Juni 1998 UML version 1.3 telah diperkaya dan direspons oleh OMG (Object Management Group), Anderson Consulting, Ericsson, Platinum Technology, ObjectTime Limited, dll serta di pelihara oleh OMG yang dipimpin oleh Cris Kobryn. UML adalah standar dunia yang dibuat oleh Object Management Group (OMG), sebuah badan yang bertugas mengeluarkan standar-standar teknologi objectoriented dan software component.

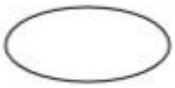
1. Use Case

Sebuah Use Case menggambarkan suatu urutan interaksi antara satu atau lebih aktor dan sistem. Dalam fase requirements, model Use Case mengambarkan

sistem sebagai sebuah kotak hitam dan interaksi antara aktor dan sistem dalam suatu bentuk naratif, yang terdiri dari input user dan respon-respon sistem. Setiap Use Case menggambarkan perilaku sejumlah aspek sistem, tanpa mengurangi struktur internalnya. Selama pembuatan model Use Case secara pararel juga harus ditetapkan obyek-obyek yang terlibat dalam setiap Use Case, untuk lebih jelasnya berikut ini adalah element - elemen pada Use Case Diagram

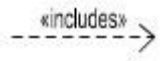


Actor : Mempresentasikan seseorang atau sesuatu(seperti perangkat,sistem lain) yang berinteraksi dengan sistem.Actor hanya berinteraksi dengan Use Case tetapi tidak memiliki kontrol atas Use Case.



Use Case : Adalah gambaran fungsionalitas dari suatu sistem, sehingga customer atau pengguna sistem paham dan mengerti mengenai kegunaan sistem yang akan dibangun.

Association : Menghubungkan link antar element.



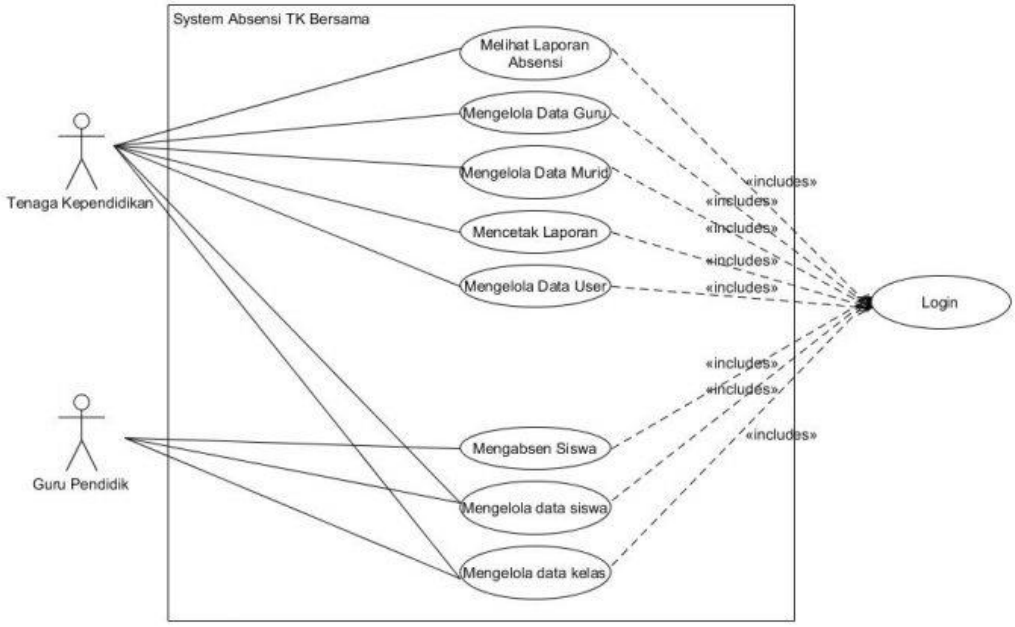
<<Include>> : Yaitu kelakuan yang harus terpenuhi agar sebuah event dapat terjadi, dimana pada kondisi ini sebuah Use Case adalah bagian dari Use Case lainnya.

Relasi Dalam Use Case

Ada beberapa relasi yang terdapat pada Use Case diagram:

1. Association, menghubungkan link antar element.
2. Generalization, disebut juga inheritance (pewarisan), sebuah elemen dapat merupakan spesialisasi dari elemen lainnya.

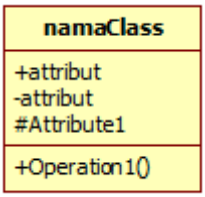
3. Dependency, sebuah element bergantung dalam beberapa cara ke element lainnya. 4. Aggregation, bentuk assosiation dimana sebuah elemen berisi elemen lainnya.



Gambar 2.4 Contoh Use Case

2. Class Diagram

Class Diagram atau Diagram Class memberikan pandangan secara luas dari suatu sistem dengan menunjukan kelas-kelasnya dan hubungan mereka. Diagram Class bersifat statis; menggambarkan hubungan apa yang terjadi bukan apa yang terjadi jika mereka berhubungan, sebuah class digambarkan seperti gambar berikut ini :



Gambar 2.5 Contoh Class

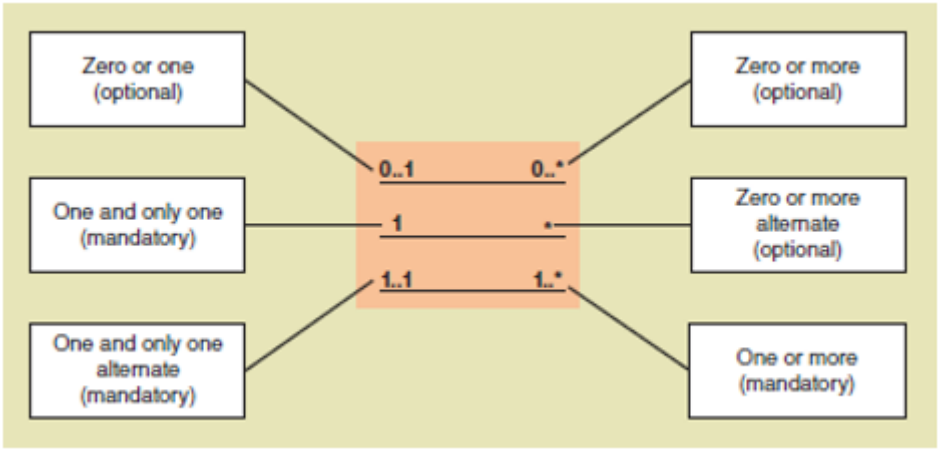
di dalam class attribut ataupun method dikenal yang namanya visibility atau access modifier, yang berguna untuk enkapsulasi komponen, ada macam macam visibility dalam diagram kelas yaitu :

1. private berarti hanya bisa diakses di class itu
2. public berarti class lain dapat mengakses attribut atau method itu
3. protected berarti hanya kelas yang diwarisi saja yang bisa mengaksesnya.

Dalam sebuah diagram kelas tentu juga ada sebuah hubungan atau relationship antara kelas, yaitu :

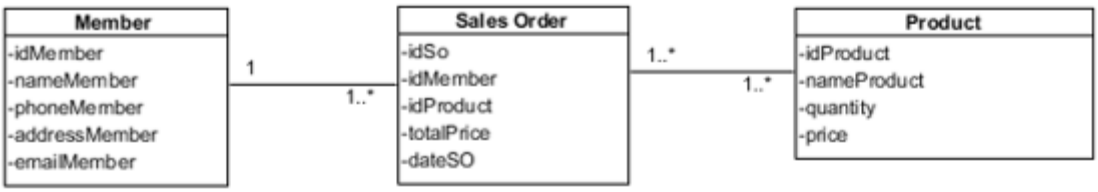
a) Asosiasi

Asosiasi, yaitu hubungan statis antar class. Umumnya menggambarkan class yang memiliki atribut berupa class lain, atau class yang harus mengetahui eksistensi class lain. Panah navigability menunjukkan arah query antar class.



Gambar 2.7 Relasi kelas

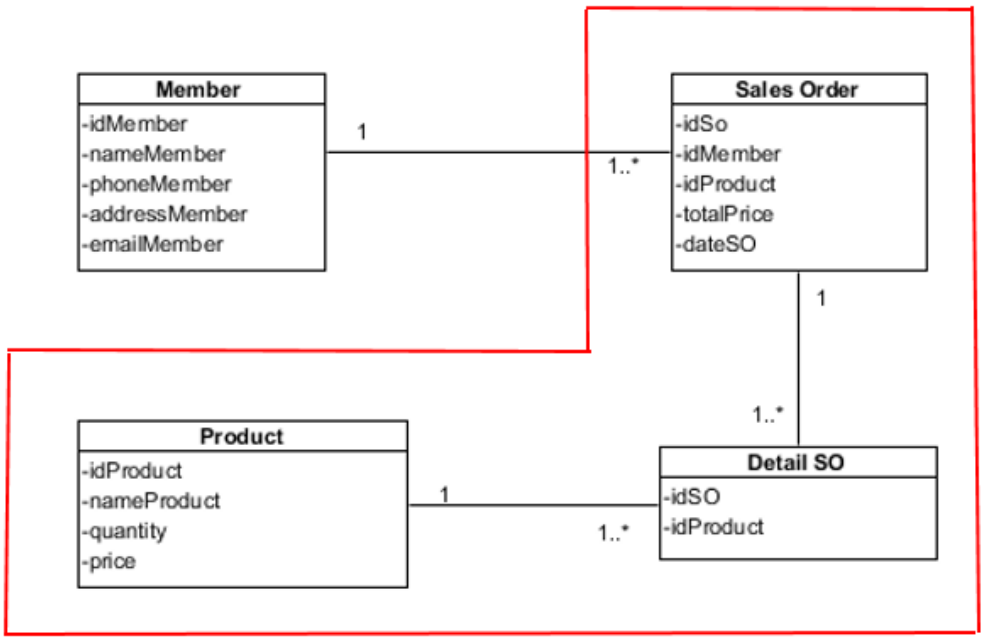
Contoh:



Gambar 2.8 domain model

Contoh diatas menunjukan domain model class yang simple dimana terdapat 3 class: Member, Sales Order, dan Product. Kita bisa lihat Member class dapat mempunyai minimal 1 dan maksimal banyak Sales Order class. Dan setiap Sales Order class dapat mempunyai minimal 1 dan maksimal 1 Member class. Sales Order class dapat mempunyai minimal 1 dan maksimal banyak Product class. Sedangkan Product class dapat mempunyai minimal 1 dan maksimal banyak Sales Order class.

Jika kita menemukan hubungan antar class seperti Sales Order dengan Product, yang disebut sebagai many-to-many asosiasi. Maka kita harus membuat Detail Class, seperti:



Gambar 2.9 Asosiasi *Many to Many*

Mengapa harus dibuat Detail Class? Apabila kita tidak membuat Detail Class maka pada Tabel database tampilannya akan seperti:

Tabel 2.5 Tabel detail class

ID sales order	ID product
S001	P001
S002	P002
S003	P003

Pada ID Sales Order S002, Product dengan ID P002 dibeli dengan jumlah 2 buah. Tetapi pada Tabel database, datanya tersimpan redudansi (double). Sehingga kita harus membuat Detail Class agar data tidak redudansi. Sehingga Sales Order class mempunyai minimal 1 dan maksimal banyak Detail SO class. Dan Detail SO class hanya mempunyai 1 Sales Order class. Sedangkan, Product class mempunyai minimal 1 dan maksimal banyak Detail SO class. Dan Detail SO class hanya mempunyai 1 Product class. Berikut ini tampilan Tabel 2.6 database pada Detail SO class:

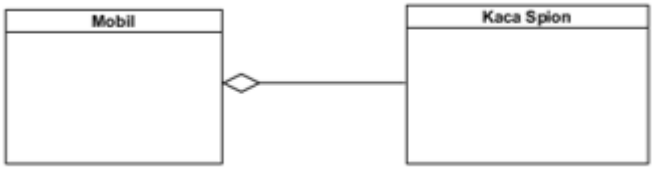
Tabel 2.6 database Detail SO

ID sales order	ID product	Quantity
S001	P001	1
S002	P002	2

Sehingga ID Sales Order dan ID Product pada Detail SO class akan sekaligus menjadi Primary Key dan Foreign Key. Dan terdapat atribut tambahan pada Detail SO class yaitu Quantity

b) Agregasi

Agregasi, yaitu hubungan antar class yang menyatakan hubungan “has-a.” Contohnya: Mobil dengan kaca spion.

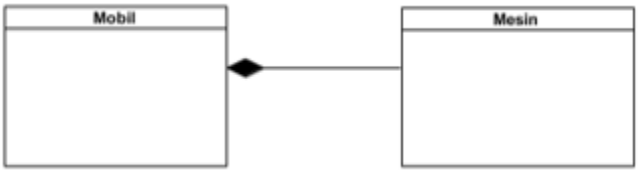


Gambar 2.10 Agregasi

Bila kita memodelkan sebuah mobil, maka dapat dikatakan bahwa mobil mempunyai kaca spion. Secara logis dapat dikatakan sebuah mobil dapat berfungsi walaupun tanpa kaca spion. Mobil dapat dipisahkan dari kaca spion

c) Komposisi

Komposisi, yaitu hubungan antar class yang menyatakan hubungan "part-of." Contohnya: Mobil dengan mesin

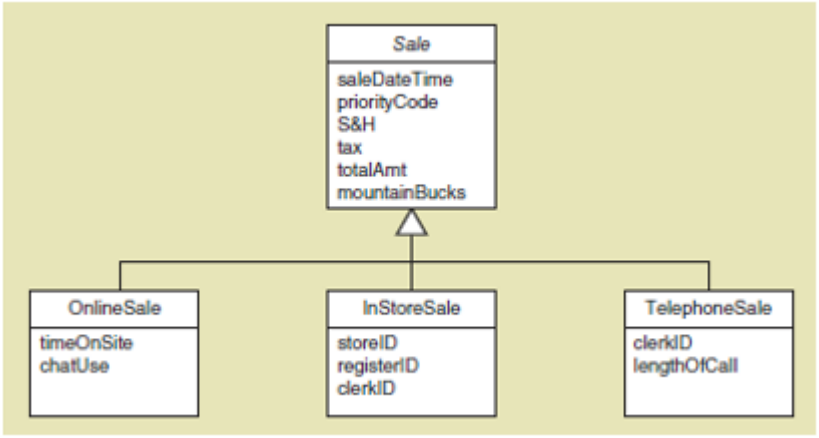


Gambar 2.11 Komposisi

Bila kita memodelkan sebuah mobil, maka dapat dikatakan bahwa mesin adalah bagian dari mobil. Secara logis dapat dikatakan bahwa mobil tidak dapat dipisahkan dari mesin. Bila mesin tidak ada disebuah mobil maka mobil itu tidak berfungsi sebagai mobil.

d) *Inheritance* dan *Generalization*

Inheritance adalah hubungan hirarkis antar class. Class dapat diturunkan dari class lain dan mewarisi semua atribut dan metoda class asalnya dan menambahkan fungsionalitas baru, sehingga ia disebut anak dari class yang diwarisinya. Kebalikan dari pewarisan adalah generalisasi.



.

Gambar 2.12 Inheritanece dan Generalitation

3. Sequence Diagram

Sequence Diagram atau Diagram sequence merupakan salah satu diagram Inter*Action* yang menjelaskan bagaimana suatu operasi itu dilakukan; message (pesan) apa yang dikirim dan kapan pelaksanaannya. Diagram ini diatur berdasarkan waktu. Obyek-obyek yang berkaitan dengan proses berjalannya operasi diurutkan dari kiri ke kanan berdasarkan waktu terjadinya dalam pesan yang terurut.

2.12 Pemodelan Analisis Terstruktur

Pemodelan analisis terstuktur merupakan teknik yang mendefinisikan spesifikasi fungsional perangkat lunak secara terstruktur melalui Data Flow Diagrams (DFD), Data Dictionary, Entity Relationship Diagram (ERD) dan State Transition Diagram (STD)

a) Diagram Konteks

Diagram konteks adalah arus data yang berfungsi untuk menggambarkan keterkaitan aliran-aliran data antara sistem dengan bagian-bagian luar (kesatuan luar). Kesatuan luar ini merupakan sumber arus data atau tujuan data yang berhubungan dengan sistem informasi tersebut. Diagram konteks memberikan batasan yang jelas mengenai besaran-besaran entitas yang berada di luar sistem

yang sedang dibuat, artinya diagram ini menggambarkan secara jelas batasanbatasan dari sebuah sistem yang sedang dibuat. Diagram konteks bisa disebut dengan "mode1 sistem pokok (Fundamental System Model) mewakili keseluruhan elemen software dengan masukan (Input) dan keluaran (Output) yang di identifikasi dengan anak panah masuk dan keluar memperlihatkan sumber".

b) Diagram Arus Data (DFD)

Data flow adalah suatu diagram yang menggunakan notasi-notasi untuk menggambarkan arus dari data sistem, yang penggunaannya sangat membantu untuk memahami sistem secara logika, terstruktur, dan jelas. Beberapa simbol yang digunakan di DFD untuk maksud mewakili :

1. Data FIow (arus data) disimbolkan dengan suatu panah. Arus data ini mengalir diantara proses (process), simpanan data (data store) dan kesatuan luar (external entity). Arus data ini menunjukan arus dari data yang dapat berupa masukan untuk sistem atau hasil dari proses sistem.
2. Proses (process), suatu proses adalah kegiatan yang dilakukan oleh orang, mesin atau komputer dari hasil suatu arus data yang masuk kedalam proses untuk dihasilkan arus data yang akan keluar dari proses.
3. Data store (simpanan data), merupakan simpanan dari data yang dapat berupa suatu file atau database, arsip atau catatan manual,agenda atau buku.

c) Kamus Data

Kamus data adalah daftar teroganisir dari semua elemen data yang ada pada suatu sistem dengan definisi yang jelas/tepat, sehingga user dan analisis sistem mendapat kesepahaman dari input, output dan komponen dari penyimpanan dan kalkulasi " intermediate" yang ada.

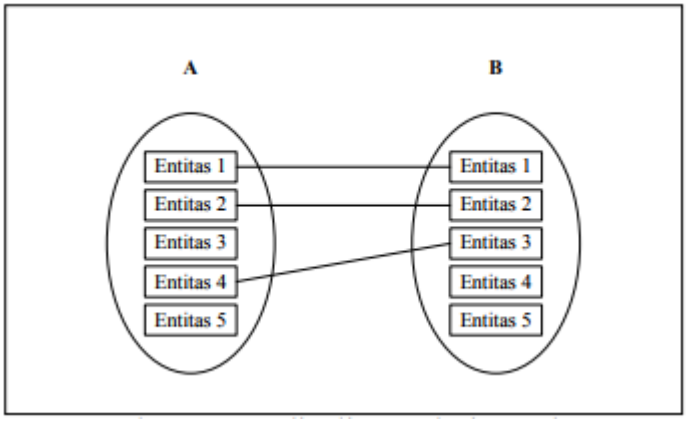
d) Desain Database

Basis data merupakan kumpulan dari data-data yang saling terkait dan saling berhubungan satu dengan lainnya. Basis data adalah kumpulan-kumpulan file yang saling berkaitan.

e) ERD (Entity Relationship Diagram)

ERD merupakan notasi grafis dalam pemodelan data konseptual yang mendeskripsikan hubungan antara penyimpanan. ERD digunakan untuk memodelkan struktur data dan hubungan antar data, karena hal ini relative kompleks. ERD menggunakan sejumlah notasi dan simbol untuk menggambarkan struktur dan hubungan antar data, simbol yang digunakan yaitu :

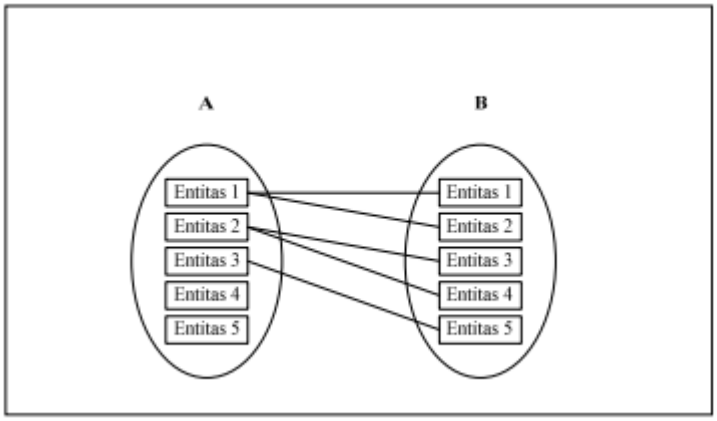
1. Entity : adalah suatu objek yang dapat diidentifikasi dalam lingkungan pemakai, sesuatu yang penting bagi pemakai dalam konteks sistem yang akan dibuat.
2. Atribut : Entiti mempunyai elemen yang disebut atribut dan berfungsi mendeskripsikan karakter entiti. Misalnya atribut nama pekerja dari entity pekerja. Setiap ERD bisa terdapat lebih dari satu atribut.
3. Hubungan : *Relationship*, sebagaimana halnya entiti maka dalam hubunganpun harus dibedakan antara hubungan atau bentuk hubungan antar entiti dengan isi dari hubungan itu sendiri. Jenis- Jenis hubungan antara lain:
   a) Satu ke Satu (*One to One*). Yang berarti setiap entitas pada himpunan entitas A berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas B, dan begitu juga sebaliknya. Setiap entitas pada himpunan entitas B berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas A



Gambar 2.13 Relasi *One to One*

b) Satu ke Banyak (*One to Many*)

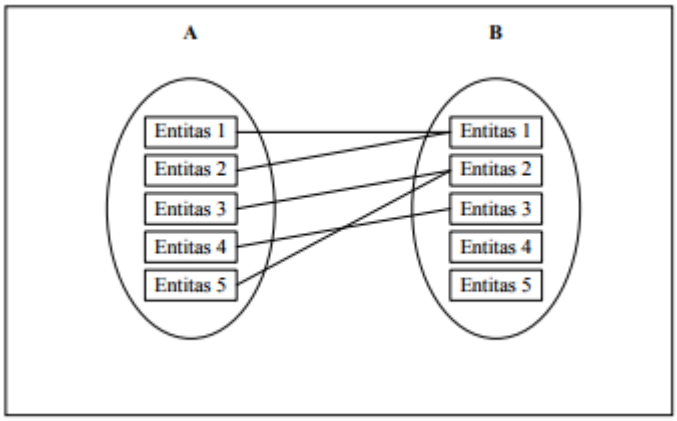
Yang berarti setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, tetapi tidak sebaliknya, dimana setiap entitas pada himpunan entitas B berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas A.



Gambar 2.14 Relasi *One to Many*

c) Banyak ke Satu (*Many to One*).

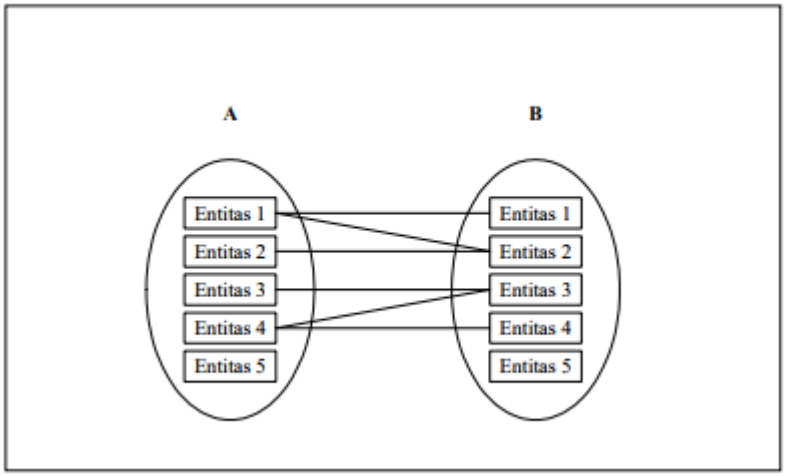
Yang berarti setiap entitas pada himpunan entitas A berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas B, tetapi tidak sebaliknya, dimana setiap entitas entitas A berhubungan dengan paling banyak satu entitas pada himpunan entitas B.



Gambar 2.15 Relasi *Many to One*

d) Banyak ke Banyak (*Many to Many*).

Yang berarti setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan B, dan demikian juga sebaliknya, dimana setiap entitas B dapat berhubungan dengan banyak entitas pada himpunan entitas.



Gambar 2.16 Relasi *Many to Many*

2.13 JAVA



Gambar 2.17 Logo JAVA 1

JAVA merupakan sebuah bahasa pemrograman dan platform komputasi yang pertama di rilis oleh Sun Microsystem pada tahun 1995, Java merupakan sebuah bahasa pemrograman yang saat ini banyak digunakan diberbagai aplikasi karena kecepatan, keamanan, dan kehandalannya, Java merupakan bahasa pemrograman tingkat tinggi yang berorientasi objek, berikut ini beberapa karakteristik dari JAVA.

1. **Sederhana**

   Bahasa pemrograman JAVA bisa dikatakan serupa dengan bahasa C atau C++, oleh karena itu para programmer yang terbiasa oleh bahasa C atau C++ tidak akan kesulitan ketika hendak mempelajari JAVA. JAVA menyederhanakan bahasa C++ dengan menambahkan beberapa fitur yang belum ada pada C++, oleh karena itu Java bisa dikatakan juga sebagai bahasa penyederhanaan dari C++ yang rumit untuk dikuasai.

2. **Object-Oritented (Berorientasi objek)**

   Bahasa yang berorientasi pada objek akan selalu memandang segala sesuatu yang ada di dunia ini merupakan sebuah objek, seperti meja, mobil, pohon dan ponsel bisa dipandang sebagai objek di JAVA, sifat object oriented akan mementingkan bagaimana objek tersebut dibuat, dimanipulasi atau digunakan, serta bagaimana objek-objek dapat saling bekerja sama. Setiap obej-objek dalam JAVA memiliki data(atribut atau properti) dan

operasi atau metoda(method), kedua hal tersebut digunakan pada saat manipulasi data.

3. Terdistribusi

JAVA dapat bekerja dengan beberapa komputer yang terdapat dalam satu jaringan(*network*) yang sama, Java memiliki sebuah pustaka kelas (Java class library) yang besar agar dapat berkomunikasi menggunakan protokol TCP/IP termasuk HTTP dan FTP.

4. *Robust* (Kuat)

Java memberikan banyak tekanan pada pengecekan awal terhadap kemungkinan kesalahan (error). Kompiler Java dapat mendeteksi lebih dini kesalahan yang mungkin muncul pada waktu program Java dijalankan (execution time). Dalam menulis sebuah program, programer juga dituntut untuk bisa menangani exception yaitu kesalahan yang muncul saat program dijalankan, sehingga program Java tetap dapat berjalan pada eksekusi normal.

5. *Architecture-Neutral* (Tidak bergantung pada platform tertentu).

Kode sumber Java (Java source code) yang sudah dirubah ke dalam bytecode Java dapat dijalankan dengan menggunakan interpreter. Keistimewaan ini menjadikan Java tidak bergantung pada platform (platform-independent). Dengan Java Virtual Machine (JVM), Anda hanya menulis program sekali dan dapat menjalankan pada platform berbeda seperti sistem operasi Microsoft Windows, OS/2, Macintosh, UNIX dan IBM AS/400.

6. *Dynamic* (Dinamis).

Tidak seperti bahasa pemrograman berorientasi obyek C++ yang memerlukan kompilasi ulang secara menyeluruh bila kelas di atasnya (parent class) mengalami perubahan, Java menggunakan metode interfes (interface) untuk menghilangkan ketergantungan ini, sehingga kelas di

bawahnya (child class) atau obyek klien (client object) tidak akan terpengaruh pada perubahan tersebut.

2.14 Android Studio

Android Studio adalah Lingkungan Pengembangan Terpadu - *Integrated Development Environment* (IDE) yang digunakan untuk pengembangan aplikasi Android, berdasarkan IntelliJ IDEA . Selain merupakan editor kode IntelliJ dan alat pengembang yang berdaya guna, Android Studio menawarkan fitur lebih banyak untuk meningkatkan produktivitas Anda saat membuat aplikasi Android, misalnya:

1. Sistem versi 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 pengujian 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

2.15 WAMP

WAMP terkadang digunakan sebagai nama singkatan untuk tumpukan perangkat lunak Windows, Apache, MySQL, PHP. Ini berasal dari LAMP yang merupakan singkatan dari Linux, Apache, MySQL, dan PHP. Seperti namanya, sementara LAMP digunakan pada server Linux, WAMP digunakan pada server Windows. Karena WordPress biasanya tidak diinstal pada Server Windows,

WAMP telah menjadi populer di kalangan pengembang sebagai metode menginstal WordPress di komputer pribadi mereka.

"A" di WAMP adalah singkatan dari Apache. Apache adalah perangkat lunak server yang digunakan untuk melayani halaman web. Setiap kali seseorang mengetikkan URL situs web WordPress Anda, Apache adalah perangkat lunak yang "melayani" situs WordPress Anda. "M" di WAMP adalah singkatan dari MySQL. MySQL adalah sistem manajemen basis data. Tugasnya di tumpukan perangkat lunak adalah menyimpan semua konten situs web Anda, profil pengguna, komentar, dll. "P" dalam WAMP adalah singkatan dari PHP. PHP adalah bahasa pemrograman yang ditulis oleh WordPress. Ini juga merupakan bagian yang menyimpan seluruh tumpukan perangkat lunak bersama. Ini berjalan sebagai proses di Apache dan berkomunikasi dengan database MySQL untuk secara dinamis membangun halaman web Anda.

Kumpulan perangkat lunak WAMP dapat diunduh dari halaman unduhan proyek wampserver. Untuk pengguna Microsoft Windows, itu datang dalam paket instalasi mudah dengan panel kontrol. Peluncuran panel kontrol manajer WAMP memulai layanan web Apache, PHP, dan MySQL di komputer lokal. WAMP dan tumpukan perangkat lunak lain dengan nama dan fitur yang mirip adalah cara yang baik untuk mengembangkan situs web di komputer lokal tanpa mentransfer file ke situs web langsung. Desainer dan pengembang tema lebih memilih untuk memiliki ini karena mempercepat waktu pengembangan mereka.

2.16 PHP



Gambar 2.18 Logo PHP 1

PHP (singkatan rekursif untuk PHP: *Hypertext Preprocessor*) adalah bahasa scripting open source dengan tujuan yang luas dan banyak digunakan yang sangat

cocok untuk pengembangan Web dan dapat dimasukkan ke dalam HTML. PHP seperti yang dikenal saat ini sebenarnya adalah penerus suatu produk bernama PHP FI. Dibuat pada tahun 1994 oleh Rasmus Lerdorf, inkarnasi pertama dari PHP adalah sekumpulan binari Common Gateway Interface (CGI) yang ditulis dalam bahasa pemrograman C. Awalnya digunakan untuk melacak kunjungan ke resume online-nya, ia menamai rangkaian skrip "*Personal Home Page Tools*," lebih sering direferensikan sebagai "*PHP Tools*." Seiring waktu, lebih banyak fungsi yang diinginkan, dan Rasmus menulis ulang *PHP Tools*, menghasilkan implementasi yang jauh lebih besar dan lebih kaya. Model baru ini mampu berinteraksi dengan basis data dan lebih banyak lagi, menyediakan kerangka kerja yang dapat digunakan pengguna untuk mengembangkan aplikasi Web dinamis sederhana seperti buku tamu.

Pada bulan Juni 1995, Rasmus merilis *Source Code* untuk *PHP Tools* ke publik, yang memungkinkan pengembang untuk menggunakannya sesuai keinginan mereka. Ini juga mengizinkan dan mendorong pengguna untuk menyediakan perbaikan untuk bug dalam kode, dan untuk meningkatkannya secara umum, berikut ini contoh *source code* bahasa PHP :

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Example</title>
  </head>
  <body>

    <?php
      echo "Hi, I'm a PHP script!";
    ?>

  </body>
</html>
```