BAB II
LANDASAN TEORI

2.1 *Event Organizer (EO)*

merupakan istilah asing yang biasa kita pakai untuk menggatikan istilah kepanitiaan atau penyelenggara acara. Menurut kata penyusunnya, *event organizer* terdiri dari kata *event* yang berarti suatu kegiatan atau acara dan *organizer* yang berarti pengorganisasi atau pengelola. Dalam bahasa yang sederhana organisasi sendiri dapat dikatakan sebagai wadah tempat berinteraksinya orang-orang untuk melakukan sesuatu demi mencapai tujuan bersama. Dalam konteks EO tujuan yang dimaksud adalah tercapainya penyelenggarakan acara tertentu dengan sukses sesuai yang telah direncanakan sebelumnya.

Sebagai pengelola suatu kegiatan, EO bertujuan agar kegiatan yang diselenggarakan dapat membawa manfaat serta keuntungan bagi pihak-pihak yang terlibat, baik bagi peserta, pembicara dan undangan yang hadir, donatur dan sponsor, maupun bagi penyelenggara sendiri. Keuntungan yang dimaksud tidak semua bersifat material, namun juga bisa bersifat non-material. Contohnya saja untuk kegiatan dengan tajuk "Mahasiswa Turun Desa". Kegiatan seperti ini tujuannya adalah pengabdian mahasiswa kepada masyarakat, tujuan yang ingin dicapai dari kegiatan ini lebih kepada upaya berbagi wawasan dari mahasiswa ke masyarakat dan dari masyarakat ke mahasiswa. Untuk sponsor dan donatur yang terlibat, umumnya juga merupakan bagian dari CSR (*Corporate Social Responsibility*) yang memang bergerak di bidang sosial.

Menjadi penyelenggara atau pengelola dari sebuah kegiatan, kadang menempatkan diri kita pada posisi dimana kita harus menyelenggarakan kegiatan sesuai dengan permintaan dari organisasi yang yang mewadahi kita.

2.2 Management *Event*

Menurut Goldblatt (Goldblatt, 2013) *Event Management* adalah kegiatan profesional mengumpulkan dan mempertemukan sekelompok orang untuk tujuan perayaan, pendidikan, pemasaran dan reuni, serta bertanggung jawab mengadakan

penelitian, membuat desain kegiatan, melakukan perencanaan dan melaksanakan koordinasi serta pengawasan untuk merealisasikan kehadiran sebuah kegiatan. Menyelenggarakan sebuah kegiatan merupakan sebuah kebutuhan yang tidak dapat dipungkiri bagi sebuah perusahaan komersial maupun non komersial. kegiatan-kegiatan tersebut bertujuan untuk menunjang kegiatan *Public Relations* (PR) dalam menciptakan citra positif perusahaan di mata *internal stakeholder* maupun *external stakeholder*. Selain itu kegiatan-kegiatan tersebut tak dapat dilepaskan dari kegiatan *marketing communication* dari perusahaan, terutama dalam kegiatan-kegiatan *below the line*

2.3 *Mobile*

Mobile adalah kata sifat yang berarti dapat bergerak atau dapat digerakkan dengan bebas dan mudah. Namun *mobile* dapat pula diartikan sebuah benda yang berteknologi tinggi dan dapat bergerak tanpa menggunakan kabel. Contohnya seperti smartphone, PDA, dan tablet. *Mobile* juga bisa diartikan kendaraan bermotor yang dapat bergerak. *Mobile* bersifat bebas seperti air dan dapat mengalir kemanapun. *Mobile* dapat berubah dan diubah dengan mudah.Jadi ada gambaran awal tentang *mobile*. Pengertian yang dibahas disini memang umum tidak sebatas smartphone atau handphone. *Mobile* bisa saja sebuah program yang dapat menjangkau seluruh tempat bisa diakses dengan mudah dan dapat diganti kapan saja tanpa kesulitan.

2.4 Program Aplikasi

Program merupakan ekpresi, pernyataan kombinasi yang disusun dan dirangkai menjadi satu kesatuan prosedur yang berupa urutan langkah untuk menyelesaikan masalah yang diimplementasikan dengan menggunakan bahasa pemrograman, sehingga dapat dieksekusi oleh komputer. Aplikasi merupakan penerapan, menyimpan sesuatu hal, data, permasalahan pekerjaan kedalam suatu sarana atau media yang digunakan untuk menerapkan atau mengimplementasikan hal atau permasalahan tersebut sehingga berubah menjadi suatu bentuk yang baru tanpa menghilangkan nilai-nilai dasar dari hal, data, permasalahan atau pekerjaan. Jadi dalam hal ini hanya bentuk dari tampilan data yang berubah, sedangkan isi yang memuat dalam data tersebut tidak mengalami perubahan. Berdasarkan pengertian dari program dan aplikasi diatas maka dapat disimpulkan, program aplikasi adalah

sederetan kode yang digunakan untuk mengatur komputer agar dapat melakukan pekerjaan sesuai dengan keinginan dari permasalahan pengguna (Vicky, 2013).

2.5 Perangkat Lunak

Software adalah perangkat lunak. Karena disebut juga sebagai perangkat lunak, maka sifatnya pun berbeda dengan hardware atau perangkat keras, jika perangkat keras adalah komponen yang nyata yang dapat diliat dan disentuh oleh secara langsung manusia, maka software atau Perangkat lunak tidak dapat disentuh dan dilihat secara fisik, software memang tidak tampak secara fisik dan tidak berwujud benda namun bisa untuk dioperasikan. Pengertian perangkat lunak komputer adalah sekumpulan data elektronik yang disimpan dan diatur oleh komputer, data elektronik yang disimpan oleh komputer itu dapat berupa program atau instruksi yang akan menjalankan suatu perintah. (Vicky, 2013)

2.6 Sistem

Pada dasarnya, sistem adalah sekumpulan elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai suatu tujuan. Sebagai gambaran, jika sebuah sistem terdapat elemen yang tidak memberikan manfaatdalam mencapai tujuan yang sama, maka elemen tersebut dapat dipastikan bukanlah bagian dari sistem (Kadir, 2014).

2.7 Informasi

McFadden, dan kawan-kawan mendefinisikan informasi sebagai data yang telah diproses sedemikian rupa sehingga meningkatkan pengetahuan seseorang yang menggunakan data tersebut. Shannon dan Weaver, dua orang insinyur listrik melakukan pendektan secara matematis untuk mendefinisikan informasi (Kroenke). Menurut mereka, informasi adalah jumlah ketidakpasian yang dikurangi ketika sebuah pesan diterima. Artinya, dengan adanya informasi, tingkat kepastian menjadi meningkat. Menurut Davis, informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau saat mendatang (Kadir, 2014).

2.8 Sistem Informasi

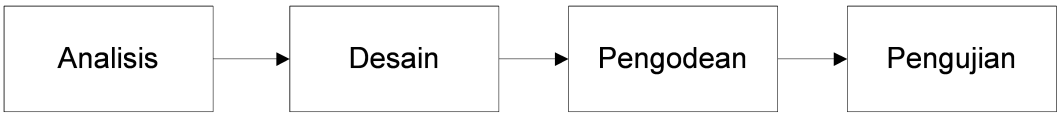
Sesungguhnya yang dimaksud sistem informasi tidak harus melibatkan komputer. Sistem informasi yang menggunakan komputer biasa disebut sistem informasi berbasis komputer (*Computer Based Information System* atau CBIS). Dalam praktik, istilah sistem informasi lebih sering dipakai tanpa embel-embel berbasis komputer, walaupun dalam kenyataannya komputer merupakan bagian yang penting. Di buku ini, yang dimaksudkan dengan sistem iformasi adalah sistem informasi berbasis komputer. Ada beragam definisi sistem informasi, yaitu :

a. Alter, sistem informasi adalah kombinasi antar prosedur kerja, informasi, orang dan teknologi informasi yang diorganisasikan untuk mencapai tujuan dalam sebuah organisasi.
b. Bodnar dan Hopwoo, sistem informasi adalah kumpulan perangkat keras dan peragkat lunak yang dirancang untuk mentransformasikan data ke dalam bentuk informasi yang berguna.
c. Gelinas, Oram dan Wiggins, sistem informasi adalah suatu sistem buatan manusia yang secara umum terdiri atas sekumpulan komponen berbasis komputer dan manual yang dibuat untuk menghimpun, menyimpan dan mengelola data serta menyediakan informasi keluaran kepada para pemakai.
d. Hall, Sistem informasi adalah sebuah rangkaian prosedur formal, dimana data dikelompokkan, diproses menjadi informasi dan didistribusikan kepada para pemakai.
e. Turban, McLean dan Wetherbe, Sebuah sistem informasi mengumpulkan, memproses, mentimpan, menganalisis dan menyebarkan informasi untuk tujuam yang spesifik.
f. Wilkinson, Sistem informasi adalah kerangka kerja yang mengkoordinasikan sumber daya (manusia dan komputer) untuk mengubah masukan (*input*) menjadi keluaran (informasi) guna mencapai sasaran-sasaran perusahaan.

Berdasarkan berbagai definisi tersebut, dapat disimpulkan bahwa sistem informasi mencangkup sejumlah komponen (manusia, komputer, teknologi informasi dan prosedur kerja), ada sesuatu yang diproses (data menjadi informasi) dan dimaksudkan untuk mencapai suatu sasaran atau tujuan (Kadir, 2014).

2.9 Metode Pengembangan Sistem

Metode air terjun *(waterfall)*sering juga di sebut model sekuensial linier atau alur hidup klasik *(classic life cycle)*. Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial, berikut adalah gambar model *waterfall* menurut (sumber: Rosa, 2016) :



Gambar 2.1 metode *waterfall*

Keterangan sebagai berikut:

1. Analisis Kebutuhan Perangkat Lunak

proses pengumpulan kebutuhan dilakukan secara intensif untuk mespesifikasikan kebutuhan perangkat lunak agar dapat di pahami oleh user.

2. Desain

Fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi anta muka, dan prosedur pengodean.

3. pembuatan kode program

hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah di buat pada tahap kedua yaitu desain.

4. pengujian

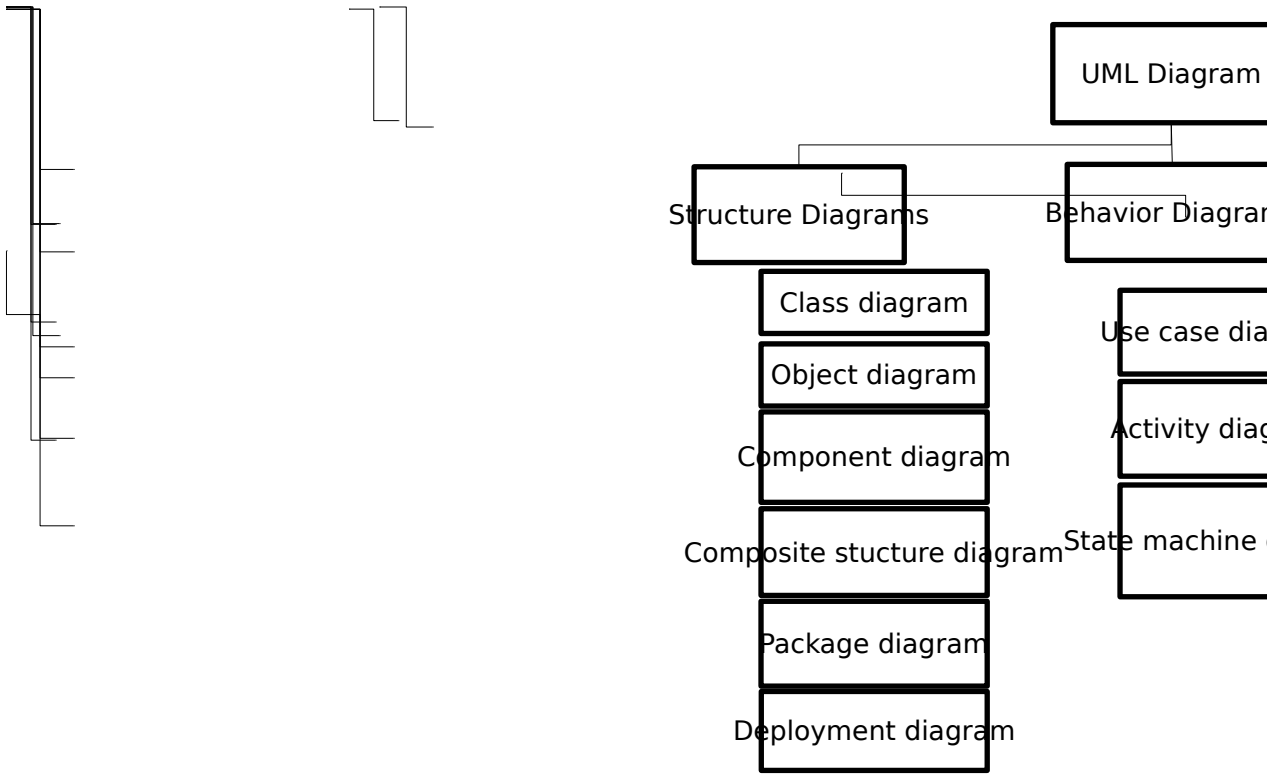
pengujian fokus pada perangkat lunak secara dari segi lojikdan fungsional dan memastikan bahwa semua bagian sudah di uji.

2.10 UML (*Unified Model Language*)

Banyak orang yang telah membuat bahasa pemodelan pembangunan perangkat lunak yang sesuai dengan teknologi pemograman yang berkembang pada saat itu, misalnya yang sempat berkembang dan digunakan oleh banyak pihak adalah *Data*

Flow Diagram (DFD) untuk memodelkan perangkat lunak yang menggunakan pemograman prosedural atau struktural, kemudian juga ada *State Transition Diagram* (STD) yang digunakan untuk memodelkan sistem *real time* (wakyu nyata).

Pada perkembangan teknik pemograman berorientasi objek, munculah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemograman berorientasi objek, yaitu *Unified Modeling Language* (UML). UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun dan mendokumentasi dari sistem perangkat lunak. UML terdiri dari 13 macam diagram yang dikelompokkan dalam tiga kategori. (Rosa, 2016).



. Gambar 2.2 Diagram UML (Sumber : Rosa, 2016)

Penjelasan dari pembagian kategori tersebut adalah :

a. *Structure diagram*, yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
b. *Behavior diagram*, yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.
c. *Interaction diagram,* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar sub sistem pada suatu sistem.

2.10.1 Use Case Diagram

Use case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami (Rosa, 2016). Ada dua hal utama pada *use case* yaitu pendefinisian apa yang dibuat aktor dan *use case*.

a. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi, walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
b. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

Tabel 2.1 Simbol *Use Case* Diagram

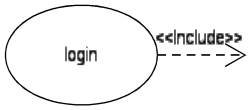
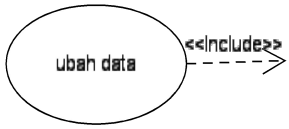
Keterangan	**Simbol**	**Deskripsi**
Use Case		Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja diawal-awal frase nama *use case*
Aktor		Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar itu sendiri. Jadi walaupun

		simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda diawal frase nama aktor.

Tabel 2.1 Simbol *Use Case* Diagram (Lanjutan)

Keterangan	Simbol	Deskripsi
Asosiasi		Komunikasi antara aktor dan *use case* yang berpartisipasi pada *use case* atau *use case* memiliki interaksi dengan aktor.
Ekstensi	«extends»	Relasi use case tambahan ke sebuah *use case*, dimana *use case* yang ditambahkan dapat berdiri sendiri walau tanpa *use case* tambahan itu; mirip dengan prinsip *inheritance* pada pemograman berorientasi objek; biasanya *use case* tambahan memiliki nama depan yang sama dengan use case yang ditambahkan, misal Arah panah mengarah pada *use case* yang ditambahkan.
Generalisasi		Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah *use case* dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya : Arah panah mengarah pada *use case* yang menjadi generalisasinya (umum).

Tabel 2.1 Simbol *Use Case* Diagram (Lanjutan)

Keterangan	Simbol	Deskripsi
Menggunakan/*include/uses*	«uses»	Ada dua sudut pandang yang cukup besar mengenai *include* di *use case* : a. Include berarti use case yang ditambahkan akan selalu dipanggil saat use case tambahan dijalankan, misal pada kasus berikut :  b. Include berarti use case yang tambahan akan selalu melakukan pengecekan apakah use case yang ditambahkan telah dijalankan sebelum use case tambahan dijalankan, misal pada kasus berikut :  Ke dua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.

2.10.2 Activity Diagram

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapta dilakukan oleh sistem. Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut :

a. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.

b. Urutan atau pengelompokan tampilan dari sistem/*user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
c. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.

Tabel 2.2 Simbol Diagram Aktivitas

Keterangan	**Simbol**	**Deskripsi**
Status awal		Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas		Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan		Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
Penggabungan		Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
Swimlane	Name	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.
Status akhir		Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

2.10.3 Squence Diagram

Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikanwaktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Banyaknya diagram sekuen yang harus digambar adalah sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interkasi jalannya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak (Rosa, 2016).

Tabel 2.3 Simbol *Squence Diagram*

Simbol	Deskripsi
Aktor	Orang, proses atau sistem lain yang bernteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang dibuat itu sendiri. Jadi, walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
Garis hidup	Men yatakan kehidupan suatu objek.
Objek	Menyatakan objek yang berinteraksi pesan.
Waktu aktif	Menyatakan objek dalam keadaan aktif dan berinteraksi pesan.
Pesan tipe *create*	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
Pesan tipe *call*	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri. Arah panah mengarah pada objek yang memiliki operasi atau metode karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.

Tabel 2.3 Simbol *Squence Diagram* (Lanjutan)

Simbol	Deskripsi
Pesan tipe *send*	Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirimi.
Pesan tipe *return*	Menyatakan bahwa suatu objek yang telah menjalankan

	suatu operasi atau metode yang menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.

2.10.4 Class Diagram

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan variabel-variabel yang dimiliki suatu kelas, sedangkan operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas (Rosa, 2016).

Kelas-kelas yang ada pada struktur sistem, harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem. Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut :

a. Kelas main
   Kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.
b. Kelas yang menangani tampilan sistem
   Kelas yang mendefinisikan dan mengatur tampilan ke pemakai.
c. Kelas yang diambil dari pendefinisian *use case*
   Kelas yang menangani fungsi-fungsi yang baru ada diambil dari pendefinisian *use case.*
d. Kelas yang diambil dari pendefinisian data
   Kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data.

Jenis-jenis kelas tersebut juga dapat digabungkan satu sama lain sesuai dengan pertimbangan yang dianggap baik asalkan fungsi-fungsi yang sebaiknya ada pada struktur kelas tetap ada. Susunan kelas juga dapat ditambahkan kelas utilitas seperti koneksi ke basis data, membaca *file* teks dan lainnya.

Dalam mengidentifikasikan metode yang ada di dalam kelas perlu memperhatikan apa yang disebut dengan *cohesion* dan *coupling*. *Cohesion* adalah ukuran seberapa dekat keterkaitan instruksi di dalam sebuah metode terkait satu sama lain, sedangkan *coupling* adalah ukuran seberapa dekat keterkaitan instruksi antara metode yang satu dengan metode yang lain dalam sebuah kelas. Sebagai aturan secara umum, maka sebuah metode yang dibuat

harus memiliki kadar *cohesion* yang kuat dan kadar *coupling* yang lemah. Simbol-simbol yang ada pada diagram kelas adalah seperti pada Tabel 2.4.

Tabel 2.4 Simbol *Class Diagram*

Simbol	Deskripsi
Kelas	Kelas pada struktur sistem.
Natarmuka/*interface*	Sama dengan konsep *interface* dalam pemograman berorientasi objek.
Asosiasi	Relasi antar kelas dalam makna umum, asosiasi biasanya juga disertai dengan *multiplicity*.
Asosiasi berarah	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan *multiplicity*.
Generalisasi	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus).

Tabel 2.4 Simbol *Class Diagram* (Lanjutan)

Simbol	Deskripsi
Kebergantungan	Relasi antar kelas dengan makna kebergantungan antar kelas.
Agregasi	Relasi antar kelas dengan maknasemua bagian (*whole-part*).

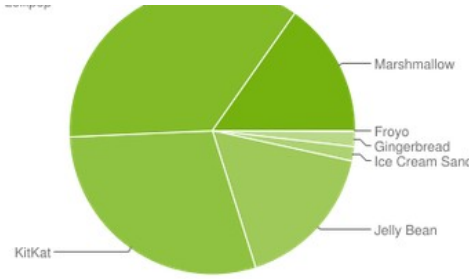
2.11 Android

Awalnya, *Android* dikembangkan oleh perusahaan kecil di Silicon Valley yang bernama *Android Inc*. Selanjutnya, *Google* mengambil alih sistem operasi tersebut pada tahun 2005 dan mencanangkannya sebagai sistem operasi yang bersifat

"*Open Source*". Sebagai konsekuensinya, siapapun boleh memanfaatkannya dengan gratis, termasuk dalam hal kode sumber yang digunakan untuk menyusun sistem operasi tersebut.

Android terus berkembang dan hal itu ditandai dengan versinya. Versi pertama hingga sekarang, dapat dilihat pada

Version	Codename	API	Distribution
2.2	Froyo	8	0.1%
2.3.3 - 2.3.7	Gingerbread	10	1.7%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	1.6%
4.1.x	Jelly Bean	16	6.0%
4.2.x		17	8.3%
4.3		18	2.4%
4.4	KitKat	19	29.2%
5.0	Lollipop	21	14.1%
5.1		22	21.4%
6.0	Marshmallow	23	15.2%



Gambar 2.3 Versi Android (sumber : https://developer.android.com)

Level API menyatakan suatu bilangan unik yang digunakan untuk mengidentifikasi API (*Application Progrmming Interface*) yang digunakan pada suatu versi Android. Dengan kata lain, setiap versi Android ditandai dengan sebuah level API (Abdul Kadir, 2013).

2.12 Java

Java menurut definisi dari Sun Microsystem adalah nama untuk sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer stand alone ataupun pada lingkungan jaringan. Java 2 adalah generasi kedua dari java platform. (Rosa, 2016).

2.13 My Sql

Menurut Solichin (2016), MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (*Database Management System*) atau DBMS yang

multithread, multi-user, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis di bawah lisensi GNU *General Public License* (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL.

2.14 Android Studio

Menurut Eric (2016), dalam jurnalnya tertulis bahwa *Android Studio* merupakan sebuah *Integrated Development Environment* (IDE) untuk *platform Android*. *Android Studio* ini diumumkan pada tanggal 16 Mei 2013 pada Konferensi Google I/O oleh Produk Manajer Google, Ellie Powers. *Android Studio* bersifat *free* dibawah *Apache License* 2.0. *Android studio* awalnya dimulai dengan versi 0.1 pada bulan mei 2013, Kemudian dibuat versi *beta* 0.8 yang dirilis pada bulan juni 2014. Yang paling stabil dirilis pada bulan Desember 2014, dimulai dari versi 1.0. Berbasiskan *JetBrainns' IntelliJ IDEA*, Studio didesain khusus untuk Android Development yang kini sudah bisa di *download* untuk *Windows, Mac OS X,* dan *Linux*.

2.15 Penelitian Terkait

Penelitian yang terkait dengan penelitian yang sudah dilakukan sebelumnya adalah sebagai berikut :

1. APLIKASI *EVENT* ORGANIZER DI KOTA SIDOARJO BERBASIS SMS GATEWAY (STUDY KASUS EO MEKS PRO ENTERPRISE) Oleh : Akhmad Husien Hamady, Suprianto, S.Si, M.Si
2. LAYANAN INFORMASI LOKASI *EVENT* MENGGUNAKAN LBS BERBASIS JQUERY *MOBILE*
   Aman Fitrullah Ardiansyah Program Studi Teknik Informatika Fakultas Teknologi Industri, Universitas Ahmad Dahlan
3. APLIKASI PERIKLANAN DAN PEMESANAN TIKET *EVENT* ORGANIZER BERBASIS WEB (2016)
   Taufiq Iqbal Fadillah, Guntur Prabawa Kusuma, Hadi Prasetyo Utomo
   Universitas Telkom (2015)