

## **BAB II**

### **LANDASAN TEORI**

#### **2.1. Pengertian Sistem**

Beberapa definisi sistem dikemukakan oleh para ahli antara lain menurut Richard F. Neuschel sebagai berikut:

“Suatu sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu”. [1]

Sedangkan definisi sistem Menurut Moscovice yang dikutip oleh Zaki Baridwan adalah:

‘Sistem adalah suatu entity (kesatuan) yang terdiri dari bagian yang saling berhubungan (subsistem) yang bertujuan untuk mencapai tujuan-tujuan tertentu’. [2]

La Midjan yang dikutip oleh Azhar Susanto memberikan definisi tentang sistem sebagai berikut:

‘Sistem sebagai kumpulan/group dari bagian/komponen apapun baik fisik maupun non fisik yang saling berhubungan satu sama lain dan berkerja sama secara harmonis untuk mencapai satu tujuan tertentu’. [3]

#### **2.2. Definisi Informasi**

Robert N. Anthony dan John Dearden menyebutkan keadaan dari sistem dalam hubungannya dengan keberakhirannya dengan istilah *entropy*. Informasi yang berguna bagi sistem akan menghindari proses *entropy* yang disebut dengan *negative entropy* atau *negentropy*. [4]

Apakah sebenarnya informasi itu, sehingga sangat penting artinya bagi suatu sistem? Informasi dapat didefinisikan sebagai berikut:

Informasi adalah data yang diolah menjadi bentuk yang lebih baik berguna dan lebih berarti bagi yang menerimanya. [5]

Sumber dari informasi adalah data. Data merupakan bentuk jamak dari bentuk tunggal data-item. Data adalah kenyataan yang menggambarkan suatu kejadian - kejadian dan kesatuan yang nyata.

### 2.3. Sistem Informasi Geografis

Sistem Informasi Geografis (SIG) pertama pada tahun 1960 yang bertujuan untuk menyelesaikan permasalahan geografis. [6]

Ada dua jenis data dalam SIG yaitu :

1. Data geografil (spatial) yaitu data yang terdiri dari lokasi eksplisit suatu geografi yang diset ke dalam bentuk koordinat.
2. Data atribut (aspatial) yaitu gambaran data yang terdiri dari informasi yang relevan terhadap suatu lokasi.

GIS merupakan akronim dari:

#### a. Geography

Istilah ini digunakan karena GIS dibangun berdasarkan pada “geografi” atau “spasial”. Object ini mengarah pada spesifikasi lokasi dalam suatu space. Objek bisa berupa fisik, budaya atau ekonomi alamiah. Penampakan tersebut ditampilkan pada suatu peta untuk memberikan gambaran yang representatif dari spasial suatu objek sesuai dengan kenyataannya di bumi.

#### b. Information

Informasi berasal dari pengolahan sejumlah data. Dalam GIS informasi memiliki volume terbesar. Setiap object geografi memiliki setting data tersendiri karena tidak sepenuhnya data yang ada dapat terwakili dalam peta. Jadi, semua data harus diasosiasikan dengan objek spasial yang dapat membuat peta menjadi intelligent. Ketika data tersebut diasosiasikan dengan permukaan geografis yang representatif, data tersebut mampu memberikan informasi dengan hanya mengklik mouse pada objek. Perlu diingat bahwa semua informasi adalah data tapi tidak semua data merupakan informasi.

### c. System

Pengertian suatu sistem adalah kumpulan elemen-elemen yang saling berintegrasi dan ber-interdependensi dalam lingkungan yang dinamis untuk mencapai tujuan tertentu.

Dengan demikian, GIS diharapkan mampu memberikan kemudahan kemudahan yang diinginkan, yaitu:

1. Penanganan data geospasial menjadi lebih baik dalam format baku
2. Revisi dan pemutakhiran data menjadi lebih mudah
3. Data geospasial dan informasi menjadi lebih mudah dicari, dianalisis dan direpresentasikan
4. Menjadi produk yang mempunyai nilai tambah
5. Kemampuan menukar data geospasial
6. Penghematan waktu dan biaya
7. Keputusan yang diambil menjadi lebih baik.

### Karakteristik SIG

- a. Merupakan suatu sistem hasil pengembangan perangkat keras dan perangkat lunak untuk tujuan pemetaan, sehingga fakta wilayah dapat disajikan dalam satu system berbasis komputer.
- b. Melibatkan ahli geografi, informatika dan komputer, serta aplikasi terkait.
- b. Masalah dalam pengembangan meliputi: cakupan, kualitas dan standar data, struktur, model dan visualisasi data, koordinasi kelembagaan dan etika, pendidikan, expert system dan decision support system serta penerapannya
- c. Perbedaannya dengan Sistem Informasi lainnya: data dikaitkan dengan letak geografis dan terdiri dari data tekstual maupun grafik
- d. Bukan hanya sekedar merupakan perubahan peta konvensional (tradisional) ke bentuk peta digital untuk kemudian disajikan (dicetak / diperbanyak) kembali
- e. Mampu mengumpulkan, menyimpan, mentransformasikan, menampilkan, memanipulasi, memadukan dan menganalisis data spasial dari fenomena geografis suatu wilayah. [6]

#### **2.4. Pemetaan**

Pemetaan adalah pengelompokan suatu kumpulan wilayah yang berkaitan dengan beberapa letak geografis wilayah yang meliputi dataran tinggi, pegunungan, sumber daya dan potensi penduduk yang berpengaruh terhadap sosial kultural yang memiliki ciri khas dalam penggunaan skala yang tepat. [7]

Pengertian lain tentang pemetaan yaitu sebuah tahapan yang harus dilakukan dalam pembuatan peta. Langkah awal yang dilakukan adalah pengumpulan data, dilanjutkan dengan pengolahan data dan penyajian dalam bentuk peta. [8]

#### **2.5. Potensi**

Menurut Wiyono, potensi dapat diartikan sebagai kemampuan yang masih terpendam dan siap untuk di wujudkan dan dapat dimanfaatkan dalam kehidupan manusia itu sendiri.

Menurut Majdi Potensi adalah kemampuan yang masih bias di kembangkan lebih baik lagi, secara sederhana potensi merupakan kemampuan terpendam yang masih perlu untuk dikembangkan.

Menurut Endra K Pihadhi, potensi adalah suatu energy ataupun kekuatan yang masih belum digunakan secara optimal. Dalam hal ini potensi diartikan sebagai kekuatan yang masih terpendam yang dapat berupa kekuatan, minat, bakat, kecerdasan, dan lain-lain yang masih beum digunakan secara optimal, sehingga manfaatnya masih belum terasa. [9]

#### **2.6. Sekolah Menengah Kejuruan (SMK)**

Pendidikan menengah kejuruan adalah pendidikan pada jenjang pendidikan menengah yang mengutamakan pengembangan kemampuan siswa untuk melaksanakan jenis pekerjaan tertentu. Pendidikan menengah kejuruan mengutamakan penyiapan siswa untuk memasuki lapangan kerja serta mengembangkan sikap profesional. Sesuai dengan bentuknya, sekolah menengah kejuruan menyelenggarakan program-program pendidikan yang disesuaikan dengan jenis-jenis lapangan kerja (Peraturan Pemerintah Nomor 29 Tahun 1990).

Sekolah Menengah Kejuruan (SMK) merupakan salah satu bentuk satuan pendidikan formal yang menyelenggarakan pendidikan kejuruan pada jenjang pendidikan menengah sebagai lanjutan dari SMP, MTs, atau bentuk lain yang sederajat. Sekolah di jenjang pendidikan dan jenis kejuruan dapat bernama Sekolah Menengah Kejuruan (SMK) atau bentuk lain yang sederajat (Undang-undang Sisdiknas Nomor 20 Tahun 2003).

SMK memiliki banyak program keahlian. Program keahlian yang dilaksanakan di SMK menyesuaikan dengan kebutuhan dunia kerja yang ada. Program keahlian pada jenjang SMK juga menyesuaikan pada permintaan masyarakat dan pasar. Pendidikan kejuruan adalah pendidikan menengah yang mempersiapkan peserta didik terutama agar siap bekerja dalam bidang tertentu.

Peserta didik dapat memilih bidang keahlian yang diminati di SMK. Kurikulum SMK dibuat agar peserta didik siap untuk langsung bekerja di dunia kerja. Muatan kurikulum yang ada di SMK disusun sedemikian rupa sesuai dengan kebutuhan dunia kerja yang ada. Hal ini dilakukan agar peserta didik tidak mengalami kesulitan yang berarti ketika masuk di dunia kerja. Dengan masa studi sekitar tiga atau empat tahun, lulusan SMK diharapkan mampu untuk bekerja sesuai dengan keahlian yang telah ditekuni.

Tujuan pendidikan menengah kejuruan menurut Undang-Undang Nomor 20 Tahun 2003, terbagi menjadi tujuan umum dan tujuan khusus. Tujuan umum pendidikan menengah kejuruan adalah :

- a. Meningkatkan keimanan dan ketakwaan peserta didik kepada Tuhan Yang Maha Esa
- b. Menembangkan potensi peserta didik agar menjadi warga Negara yang berakhlak mulia, sehat, berilmu, cakap, kreatif, mandiri, demokratis dan bertanggung jawab;
- c. Mengembangkan potensi peserta didik agar memiliki wawasan kebangsaan, memahami dan menghargai keanekaragaman budaya bangsa Indonesia
- d. Mengembangkan potensi peserta didik agar memiliki kepedulian terhadap lingkungan hidup dengan secara aktif turut memelihara dan melestarikan

lingkungan hidup, serta memanfaatkan sumber daya alam dengan efektif dan efisien.

Tujuan khusus pendidikan menengah kejuruan adalah sebagai berikut:

- a) Menyiapkan peserta didik agar menjadi manusia produktif, mampu bekerja mandiri, mengisi lowongan pekerjaan yang ada sebagai tenaga kerja tingkat menengah sesuai dengan kompetensi dalam program keahlian yang dipilihnya
- b) Menyiapkan peserta didik agar mampu memilih karir, ulet dan gigih dalam berkompentensi, beradaptasi di lingkungan kerja dan mengembangkan sikap profesional dalam bidang keahlian yang diminatinya
- c) Membekali peserta didik dengan ilmu pengetahuan, teknologi dan seni agar mampu mengembangkan diri di kemudian hari baik secara mandiri maupun melalui jenjang pendidikan yang lebih tinggi
- d) Membekali peserta didik dengan kompetensi kompetensi yang sesuai dengan program keahlian yang dipilih. [10]

### **2.7. Akreditasi**

Menurut Drs.A.Fatah Munzali Akreditasi sekolah adalah ‘suatu kegiatan penilaian kelayakan suatu sekolah berdasarkan kriteria yang telah ditetapkan dan dilakukan oleh Badan Akreditasi Sekolah yang hasilnya diwujudkan dalam bentuk pengakuan peringkat kelayakan’.

Tujuan Akreditasi memperoleh gambaran kinerja sekolah yang dapat digunakan sebagai alat pembinaan, pengembangan dan peningkatan mutu pendidikan. Menentukan tingkat kelayakan suatu sekolah dalam penyelenggaraan pelayanan pendidikan.

Tujuan Akreditasi tersebut memiliki makna bahwa hasil akreditasi :

1. Memberikan gambaran tentang tingkat kinerja sekolah yang dapat digunakan untuk kepentingan pembinaan, pengembangan, dan peningkatan kinerja sekolah, baik kualitas , produktivitas, efektifitas, efisiensi, dan inovasinya.

2. Memberikan jaminan kepada publik bahwa sekolah tertentu yang telah dinyatakan terakreditasi menyediakan layanan pendidikan yang memenuhi standar kualitas nasional.
3. Memberikan jaminan kepada publik bahwa siswa dilayani oleh sekolah yang benar-benar memenuhi persyaratan standar kualitas nasional. [11]

### 2.8. Aplikasi Web

Halaman - halaman web yang telah ditambahkan kode program (PHP, ASP, JSP, Perl, dan lain sebagainya) biasa dikenal dengan nama *web application* (aplikasi web). Saat ini terdapat berbagai macam aplikasi web, diantaranya adalah *webmail*, *online shopping*, *blog*, *search engine* (mesin pencarian), SFA (*Sales Force Automation*), ERP (*Enterprise Resource Planning*), *online auction* (lelang online), dan CRM (*Customer Relationship Management*). Tentunya masih banyak jenis aplikasi web lainnya yang belum kita sebutkan. Aplikasi - aplikasi *web* tersebut dapat diakses lewat internet maupun intranet (misal intranet perusahaan).

Aplikasi *web* adalah aplikasi yang dapat diakses dengan menggunakan web browser lewat jaringan baik internet ataupun intranet (misalnya intranet perusahaan). Contoh aplikasi web yang banyak digunakan misalnya Yahoo Mail, GMail, Google search engine, Friendster, YouTube, Facebook, Flickr, WordPress, dan lain sebagainya. [12]

### 2.9. Arsitektur Aplikasi

Istilah *tier* saat ini digunakan untuk menjelaskan pembagian sebuah aplikasi yang melalui *client/server*. Pembagian proses kerja adalah bagian utama dari konsep *client/server* saat ini. Jadi saat ini pembagian kerja pada *client* dan *server* telah diatur secara lebih spesifik.

1. *2-tier* – membagi proses load ke dalam dua bagian. Aplikasi utama secara logika dijalankan pada sisi *client* yang biasanya mengirimkan *request* dalam bentuk sintaks SQL ke sebuah *database server* yang berfungsi sebagai media penyimpanan data atau dapat juga disebut arsitektur fat *client* karena bagian terbesar atau yang utama dari aplikasi, berjalan pada sisi *client/komputer client*.

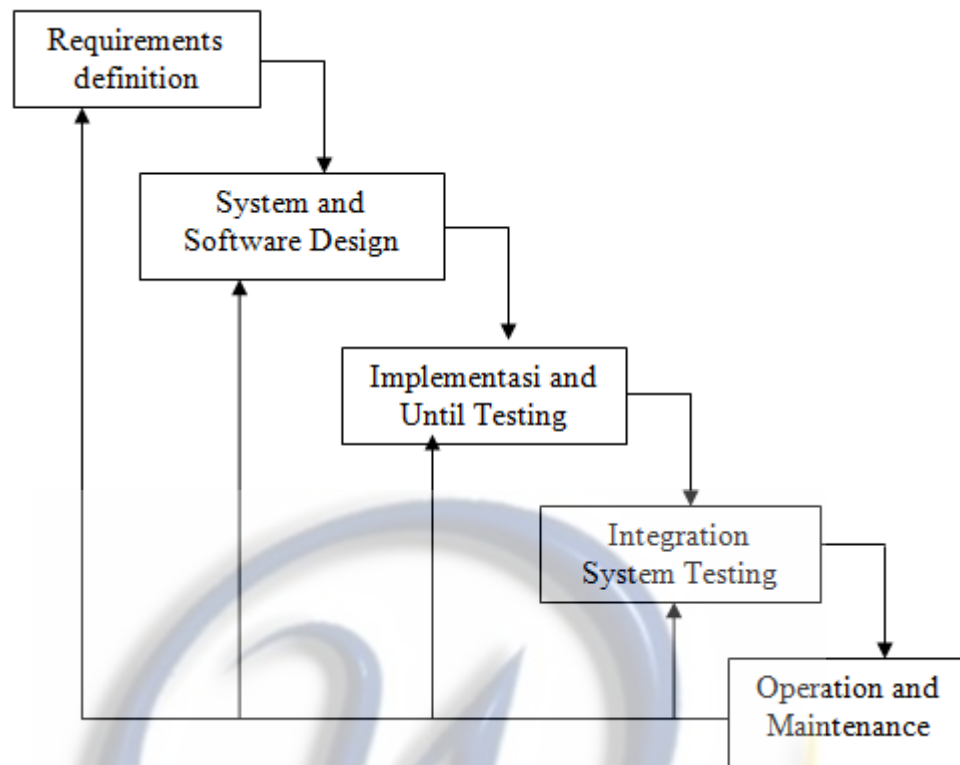
2. *3-tier* – membagi proses loading antara lain:
  - a. Komputer *client* menjalankan *GUI logic*.
  - b. Aplikasi *server* menjalankan *business logic*.
  - c. *Database* dan/ atau *legacy application*.

Karena *3-tier* ini memindahkan *application logic* ke *server* sehingga sering juga disebut arsitektur *fat server*. [13]

### 2.10. Metode Waterfall

Model ini mengambil kegiatan proses dasar seperti spesifikasi, pengembangan, validasi, dan evolusi, dan merepresentasikannya sebagai fase-fase proses yang berbeda seperti spesifikasi persyaratan, perancangan perangkat lunak, implementasi, pengujian, dan seterusnya.

Model pertama yang diterbitkan untuk proses pengembangan perangkat lunak diambil dari proses rekayasa lain (Royce, 1970). Berkat penurunan dari satu fase ke fase yang lainnya, model ini dikenal sebagai ‘model air terjun’ atau siklus hidup perangkat lunak. Tahap-tahap utama dari model ini memetakan kegiatan-kegiatan pengembangan dasar yaitu:



Gambar 0-1 Sketsa metode waterfall [14]

**1. Requirement definition**

Pelayanan, batasan, dan tujuan sistem ditentukan melalui konsultasi dengan user sistem. Persyaratan ini kemudian didefinisikan secara rinci dan berfungsi sebagai spesifikasi sistem.

**2. System and Software Design**

Proses perancangan sistem membagi persyaratan dalam sistem perangkat keras atau perangkat lunak. Kegiatan ini menentukan arsitektur sistem secara keseluruhan. Perancangan perangkat lunak melibatkan identifikasi dan deskripsi abstraksi sistem perangkat lunak yang mendasar dan hubungan-hubungannya.

**3. Implementasi and Until Testing**

Pada tahap ini, perancangan perangkat lunak direalisasikan sebagai serangkaian program atau unit program. Pengujian unit melibatkan verifikasi bahwa setiap unit telah memenuhi spesifikasinya.

**4. Integration System Testing**

Unit program atau program individual diintegrasikan dan diuji sebagai sistem yang lengkap untuk menjamin bahwa persyaratan sistem telah dipenuhi. Setelah pengujian sistem, perangkat lunak dikirim kepada pelanggan.

#### 5. *Operation and Maintenance*

Biasanya (walaupun tidak seharusnya), ini merupakan fase siklus hidup yang paling lama. Sistem diinstal dan dipakai. Pemeliharaan mencakup koreksi dari berbagai error yang tidak ditemukan pada tahap-tahap terdahulu, perbaikan atas implementasi unit sistem dan pengembangan pelayanan sistem, sementara persyaratan-persyaratan baru ditambahkan. [14]

### 2.11. Konsep Berorientasi Objek

Dalam konsep berorientasi objek, abstraksi dilakukan pada level objek. Objek adalah representasi dunia nyata kedalam bahasa pemrograman. Objek bisa dalam bentuk benda nyata seperti manusia, hewan, kendaraan. Di dalam pemrograman berorientasi objek, kode dan data yang saling terkait dibungkus dalam satu entitas.

Ada 3 prinsip utama berorientasi objek, yaitu:

#### 1. Enkapsulasi

Enkapsulasi adalah mekanisme penggabungan kode dan data yang dimanipulasi oleh kode tersebut. Mekanisme ini juga menjaga kode dan data tetap aman dari *interferensi* dan penggunaan oleh pihak yang tidak berhak.

#### 2. Pewarisan (*inheritance*)

Inheritance adalah proses pewarisan data dan method dari suatu kelas kepada kelas yang lain. Pewarisan ini bersifat menyeluruh, sehingga semua data dan method yang dimiliki oleh kelas asalnya akan diturunkan kepada kelas yang baru. Pewarisan ini adalah konsep untuk meningkatkan guna ulang (*reuse*) perangkat lunak.

#### 3. *Polymorphism*

*Polymorphism* (dari bahasa Yunani yang berarti banyak rupa), adalah mekanisme yang memungkinkan suatu modul memiliki nama yang sama, namun memiliki *behavior* (tingkah laku) yang berbeda, sehingga implementasi dalam modulnya pun berbeda. [15]

## 2.12. Unified Modelling Language (UML)



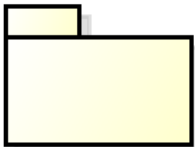
*Unified Modelling Language* (UML) bukanlah suatu proses melainkan bahasa pemodelan secara grafis untuk menspesifikasikan, memvisualisasikan, membangun, dan mendokumentasikan seluruh rancangan sistem perangkat lunak.

Penggunaan model ini bertujuan untuk mengidentifikasi bagian - bagian yang termasuk dalam lingkup sistem yang dibahas dan bagaimana hubungan antara sistem dengan subsistem maupun sistem lain di luarnya.

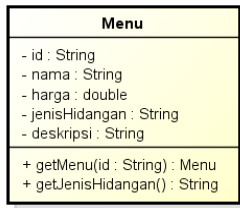



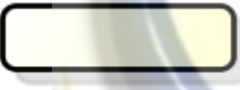
Dengan pemodelan menggunakan UML, pengembang dapat melakukan:

1. Tinjauan umum bagaimana arsitektur sistem secara keseluruhan.
2. Penelaahan bagaimana objek-objek dalam sistem saling berinteraksi, mengirimkan pesan dan saling bekerjasama satu sama lain.
3. Menguji apakah sistem perangkat lunak sudah berfungsi seperti seharusnya.
4. Dokumentasi sistem perangkat lunak untuk keperluan-keperluan tertentu dimasa yang akan datang. [15]

Tabel 0-1 Daftar simbol-simbol UML [15]

Gambar Simbol	Nama Simbol	Keterangan
	<i>Usecase</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
	<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
	<i>Package</i>	Simbol yang memberikan batasan dan komentar yang dikaitkan pada suatu elemen atau kumpulan elemen

Tabel II-1 Daftar simbol–simbol UML (Lanjutan) [15]

Gambar Simbol	Nama Simbol	Keterangan
 <pre> classDiagram     class Menu {         - id : String         - nama : String         - harga : double         - jenisHidangan : String         - deskripsi : String         + getMenu(id : String) : Menu         + getJenisHidangan() : String     } </pre>	<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
	<i>Control</i>	Mengkordinasikan aktifitas dalam sistem
	<i>Entity</i>	Kelas yang berhubungan data dan informasi yang dibutuhkan oleh sistem
	<i>Boundary</i>	Kelas yang memodelkan interaksi antar satu atau lebih aktor dengan sistem
	<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain

UML menyediakan 9 jenis diagram yang dapat dikelompokkan berdasarkan sifatnya, yaitu:

### 2.12.1. Use Case Diagram

Use case adalah deskripsi fungsi sistem dari sudut pandang pengguna. Use case diagram digunakan untuk menunjukkan fungsi yang sistem akan sediakan dan menunjukkan pengguna mana yang akan berkomunikasi dengan sistem. Use case diagram ini dikembangkan oleh Jacobson et al. (1992), dan judul buku dimana use case diagram ini ditampilkan adalah A Use Case Driven Approach (Bennett, McRobb, dan Farmer, 2006). Menurut Bennett, McRobb, dan Farmer (2006), use case diagram menunjukkan tiga aspek dalam sistem yaitu:

1. *Actor*

*Actor* mewakili peran yang dimiliki orang, sistem lain, atau *device* ketika berkomunikasi dengan *use cases* tertentu dalam sistem dimana satu *actor* dapat mewakili beberapa orang atau pekerjaan.

## 2. Use case

Sebuah use case menggambarkan sebuah fungsi yang dilakukan oleh sistem untuk mencapai tujuan pengguna dimana digambarkan dengan bentuk elips.

## 3. Batasan sistem atau subsistem





Batasan yang mengelilingi fungsi-fungsi dalam sistem atau subsistem dimana berbentuk persegi panjang.

Penggunaan model ini bertujuan untuk mengidentifikasi bagian-bagian yang termasuk dalam lingkup sistem yang dibahas dan bagaimana hubungan antara sistem dengan subsistem maupun sistem lain diluarnya.







Dengan pemodelan menggunakan *UML*, pengembang dapat melakukan:

1. Tinjauan umum bagaimana arsitektur sistem secara keseluruhan.
2. Penelaahan bagaimana objek-objek dalam sistem saling berinteraksi, mengirimkan pesan dan saling bekerjasama satu sama lain.
3. Menguji apakah sistem perangkat lunak sudah berfungsi seperti seharusnya.

Tabel 0-2 Daftar simbol use case diagram [15]

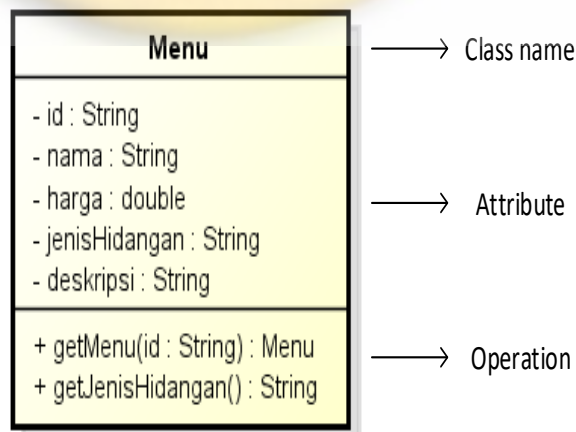
NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri ( <i>independent</i> ).
3		<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> ).
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .

Tabel II-2 Daftar simbol use case diagram (Lanjutan) [15]

5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

### 2.12.2. Class Diagram

*Class* adalah sebuah spesifikasi yang akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metode/fungsi). *Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti pewarisan, asosiasi, dan lain-lain.



Gambar 0-2 Contoh class diagram

*Class* memiliki tiga area pokok:

1. Nama (*class name*)
2. Atribut
3. Metode (*operation*)

Pada UML, *class* digambarkan dengan segi empat yang dibagi beberapa bagian. Bagian atas merupakan nama dari *class*. Bagian tengah merupakan struktur dari *class* (atribut) dan bagian bawah merupakan sifat dari *class* (metode/operasi).

Atribut dan metode dapat memiliki salah satu sifat berikut:

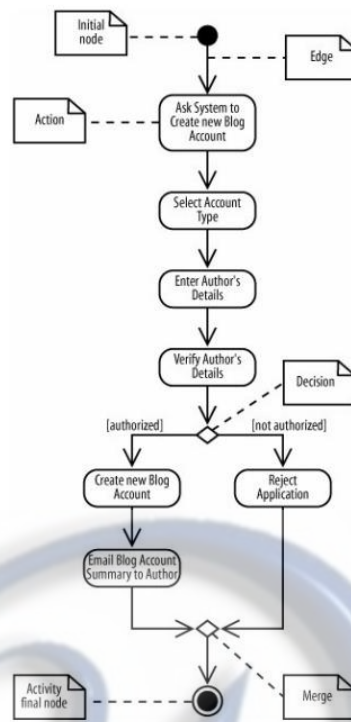
1. *Private*, tidak dapat dipanggil dari luar *class* yang bersangkutan.
2. *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan *class* lain yang mewarisinya.
3. *Public*, dapat dipanggil oleh *class* lain.

Hubungan antar *class*:

1. Asosiasi, yaitu hubungan statis antar *class*. Umumnya menggambarkan *class* yang memiliki atribut yang berupa *class* lain, atau *class* yang harus mengetahui eksistensi *class* lain.
2. Agregasi, yaitu hubungan yang menyatakan bagian (“terdiri atas”).
3. Pewarisan, yaitu hubungan hirarki antar *class*. *Class* dapat diturunkan dari *class* lain dan mewarisi semua atribut dan metode *class* asalnya serta bisa menambahkan fungsionalitas baru sehingga *class* tersebut disebut anak dari *class* yang diwarisinya.
4. Hubungan dinamis, yaitu rangkaian pesan (*message*) *class* dari satu *class* kepada *class* lain. Hubungan dinamis dapat digambarkan dengan menggunakan *sequence diagram*.

### 2.12.3. Activity Diagram

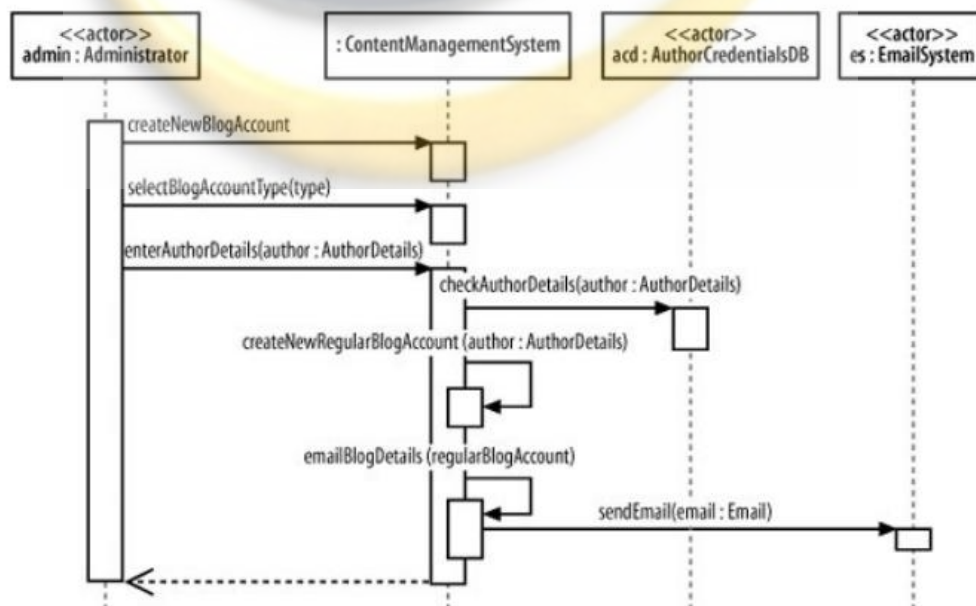
Activity Diagram adalah representasi secara grafis dari proses dan *control flow* dan berfungsi untuk memperlihatkan alur dari satu aktivitas ke aktivitas yang lain serta menggambarkan perilaku yang kompleks.



Gambar 0-3 Contoh activity diagram [16]

**2.12.4. Sequence Diagram**

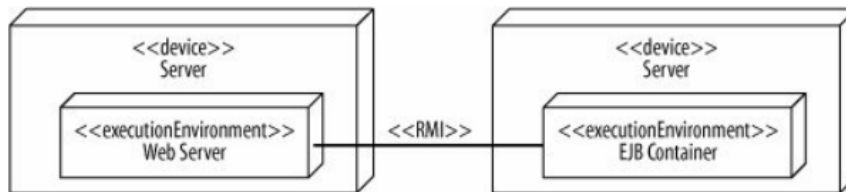
Sequence diagram adalah salah satu dari beberapa macam UML Sebuah sequence diagram menunjukkan sebuah interaksi antara objek-objek yang disusun dalam urutan waktu. Kegunaan dari sequence diagram adalah menunjukkan interaksi objek secara rinci yang terjadi untuk satu use case.



Gambar 0-4 Contoh sequence diagram [16]

### 2.12.5. Deployment Diagram

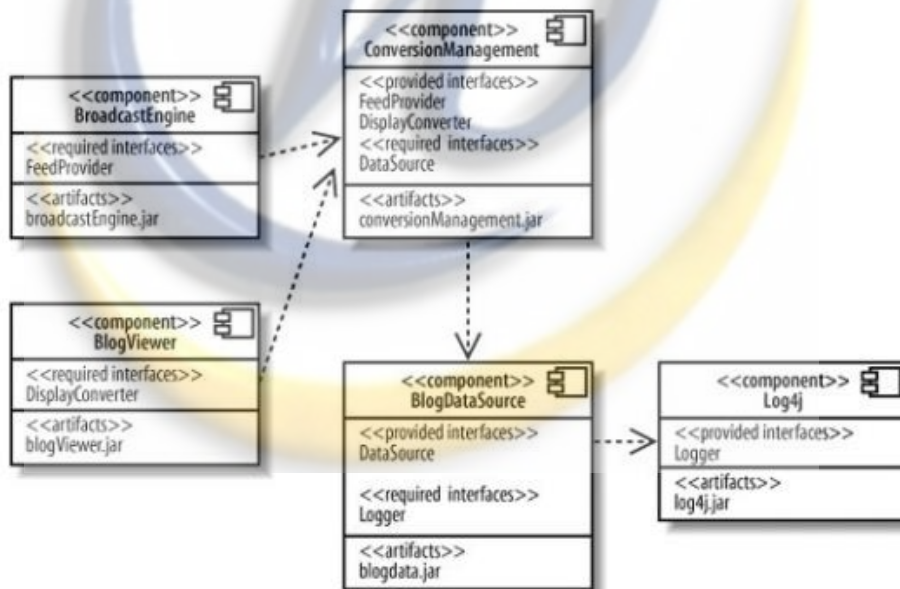
Merupakan penggambaran tugas-tugas kongkrit dari setiap *node/software* yang terlibat dalam jaringan sistem, menampilkan keseluruhan *node* dalam jaringan serta hubungan dari *node-node* tersebut termasuk proses-proses yang terlibat di dalamnya.



Gambar 0-5 Contoh deployment diagram [16]

### 2.12.6. Component Diagram

*Component diagram* menunjukkan organisasi dan ketergantungan di antara sekumpulan komponen. Diagram ini memodelkan pandangan implementasi fisik dari sistem.



Gambar 0-6 Contoh component diagram [16]

## 2.13. Teknologi

Teknologi yang digunakan dalam pembuatan aplikasi ini adalah sebagai berikut:

### 2.13.1. Internet

Menurut O'Brien dalam bukunya *Introduction to Information System* mendefinisikan internet sebagai berikut:

Sebuah jaringan besar yang terdiri dari berbagai jaringan yang meliputi jaringan bersifat bisnis, pendidikan dan riset serta menghubungkan jutaan komputer di dalam jaringan-jaringan tersebut. [17]

Ellsworth Jill H. & Ellsworth Matthew V mendefinisikan WWW sebagai berikut: WWW adalah sistem client/server yang dirancang untuk menggunakan dokumen hypertext dan hypermedia via Internet. WWW menggunakan HTTP (*Hypertext Transfer Protocol*) untuk bertukar informasi, image, dan data lain. Dokumen diformat dalam HTML (*Hypertext Markup Language*) yang digunakan untuk menciptakan halaman dan dokumen yang disajikan pada web. [18]

Ellsworth Jill H. & Ellsworth Matthew V mendefinisikan URL sebagai berikut: URL merupakan singkatan dari *Uniform Resources Locator* adalah cara standar yang digunakan untuk menentukan situs atau halaman pada internet. URL merupakan cara standar untuk menampilkan informasi tentang jenis isi dan lokasi file: nama file, lokasi komputer di internet, letak file di dalam komputer, dan protokol internet yang digunakan untuk mengakses file itu. [18]

### **2.13.2. MySQL**

MySQL adalah sebuah implementasi dari sistem manajemen basisdata relasional (RDBMS) yang didistribusikan secara gratis. Setiap pengguna dapat secara bebas menggunakan MySQL, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basisdata yang telah ada sebelumnya; SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian basisdata, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. [19]

### **2.13.3. Framework**

Framework adalah kerangka kerja. *Framework* juga dapat di artikan sebagai kumpulan script (terutama *class* dan *function* yang dapat membantu *developer/programmer* dalam menangani berbagai masalah-masalah dalam pemrograman seperti koneksi ke database, pemanggilan variabel, dan file. Sehingga *developer* lebih fokus dan lebih cepat membangun aplikasi. [20]

#### **2.13.4. MVC (Model-View-Controller)**

*Model-View-Controller* atau MVC adalah sebuah metode untuk membuat sebuah aplikasi dengan memisahkan data (*Model*) dari tampilan (*View*) dan cara bagaimana memprosesnya (*Controller*). Dalam implementasi kebanyakan framework dalam aplikasi web adalah berbasis arsitektur MVC (*Model-View-Control*). [20]

#### **2.13.5. Codeigniter**

*Codeigniter* adalah *framework php* yang berjalan pada *php 4* dan *php 5*. Tujuan utama dari *CodeIgniter* adalah untuk memudahkan *programmer* dalam mengembangkan aplikasi secara cepat tanpa harus melakukan pemrograman dari nol. [20]

#### **2.13.6. PHP (PHP: Hypertext Preprocessor)**

PHP merupakan bahasa pemrograman yang paling banyak dipakai untuk pengembangan web dan dapat ditanamkan ke dokumen HTML. Pengeksekusian kode PHP dilakukan pada *server*, menghasilkan dokumen HTML yang kemudian dikirimkan ke *client*. PHP sangat sederhana untuk para pendatang baru, akan tetapi juga menawarkan fitur-fitur canggih bagi para programmer profesional. [21]

#### **2.13.7. HTML (HTML:Hyper Text Markup Language)**

*HyperText Markup Language* (HTML) adalah bahasa dari *World Wide Web* (www) yang dipergunakan untuk menyusun dan membentuk dokumen agar dapat ditampilkan pada program web browser. HTML juga dapat disebut sebagai protocol yang digunakan untuk mentransfer data atau dokumen dari web server ke browser. HTML inilah yang menjadi dasar bila akan menjelajah internet dan melihat halaman web yang menarik. [22]

#### **2.13.8. JavaScript**

JavaScript adalah bahasa scripting kecil, ringan, berorientasi objek yang ditempelkan pada kode HTML dan di proses di sisi client. JavaScript digunakan dalam pembuatan website agar lebih interaktif dengan memberikan kemampuan tambahan terhadap HTML melalui eksekusi perintah di sisi browser. JavaScript dapat merespon perintah user dengan cepat dan menjadikan halaman web menjadi responsif. *JavaScript* memiliki struktur

sederhana, kodenya dapat disisipkan pada dokumen HTML atau berdiri sebagai satu kesatuan aplikasi. [23]

Struktur penulisan JavaScript adalah sebagai berikut.

```
<script language = "javascript">  
  
<!--Penulisan kode javascript/-->  
  
</script>
```

### 2.13.9. CSS Framework

CSS framework adalah library CSS yang digunakan untuk membuat standarisasi layout dan tema dari website. Dengan CSS framework kita bisa membuat Front-end website dengan cepat mengingat layout dan tampilan sudah disediakan. [24]

Jika kita bandingkan dengan PHP framework dan Javascript framework, maka tugas masing masing adalah sebagai berikut:

1. PHP Framework , mempercepat proses development dari logic bisnis aplikasi (back end)
2. Javascript framework mempercepat proses developmet untuk AJAX, animasi, UI dan interaksi dengan User.
3. CSS Framework bertugas membuat tampilan layout yang standar, rapih, dan tampil sempurna disetiap browser atau device

### 2.13.10. XAMPP

XAMPP adalah "*Web server dalam kotak*": Apache, MySQL, dan PHP (ditambah beberapa komponen lainnya), semua dalam satu paket, paket bebas berlisensi mudah diinstall, pra-konfigurasi. Hal ini sering digunakan sebagai *sandbox* untuk menguji perangkat lunak pada komputer desktop yang satu ini. XAMPP sering berfungsi sebagai alternatif yang lebih mudah dan lebih aman untuk menginstal perangkat lunak pada server *Web* yang sebenarnya, tetapi bisa digunakan untuk menjalankan perangkat lunak produksi. XAMPP juga menawarkan "*USB Lite*" versi (sebelumnya hanya "*Lite*" versi) dirancang untuk digunakan pada media *portabel*, yang sangat membantu untuk tujuan kita. [25]

### 2.13.11. Goggle Maps API

Menurut Shodiq yang dikutip oleh Falahah Google Maps adalah ‘layanan gratis berupa peta interaktif yang disediakan oleh Google’. Layanan ini berupa sekelompok API (Application Program Interface) yang dapat ditambahkan pada aplikasi berbasis web. Goole Maps API sendiri memuat pustaka Javascript yang memudahkan kita untuk menyisipkan layanan peta berbasis framework Google, hanya cukup dengan menuliskan kode HTML, Javascript dan koneksi internet. Adanya layanan peta gratis ini memudahkan pengembang menampilkan data secara spasial dan membuat pengembang dapat lebih berfokus pada fitur aplikasi. Langkah untuk menyisipkan Google Maps API adalah sebagai berikut: a. Sisipkan Javascript Maps API ke dalam HTML b. Tambahkan elemen div dengan nama map\_canvas untuk menampilkan peta. c. Siapkan beberapa obyek literal untuk menyimpan property pada peta d. Tuliskan fungsi Javascript untuk membuat peta e. Insialisasi peta pada tag body HTML melalui event load. [26]

