

BAB II

TINJAUAN PUSTAKA

II.1. Sistem Informasi

Menurut Eddy Prahasta (2014 : 70-78) Sistem Informasi dapat didefinisikan sebagai sekumpulan objek, ide dan data yang telah ditempatkan pada konteks yang penuh arti oleh penerimanya, berikut keterkaitannya di dalam mencapai tujuan. Dengan kata lain, sistem informasi adalah sekumpulan komponen (sub-sistem fisik & non-fisik) dan informasi yang saling berhubungan satu sama lainnya dan bekerja sama untuk mencapai suatu tujuan.

II.2. Sistem Informasi Geografis

Menurut Eddy Prahasta (2014 : 95) Pada dasarnya, istilah sistem informasi geografis (SIG) merupakan gabungan tiga unsur pokok: sistem, informasi, dan geografis. Dengan demikian, pengertian terhadap ketiga unsur pokok ini sangat membantu dalam memahami SIG. Dengan melihat unsur-unsur pokoknya, maka jelas bahwa SIG juga merupakan tipe sistem informasi seperti yang telah dibahas di muka; tetapi dengan tambahan unsur “Geografis”. Jadi, SIG merupakan sistem yang menekankan pada unsur “informasi geografis”.

Istilah “Geografis” merupakan bagian dari spasial. Istilah ini sering digunakan secara bergantian/tertukar satu sama lainnya hingga muncullah istilah yang ketiga geospasial. Ketiga istilah ini mengandung pengertian yang kurang lebih serupa di dalam konteks SIG. Penggunaan kata “Geografis”

mengandung pengertian suatu hal mengenai bumi: baik permukaan dua dimensi atau tiga dimensi. Dengan demikian, istilah “informasi geografis” mengandung pengertian informasi mengenai tempat-tempat yang terletak di permukaan bumi, pengetahuan mengenai posisi dimana suatu objek terletak di bumi, atau informasi mengenai keterangan objek yang terdapat di permukaan bumi yang posisinya diketahui. Dengan pengertian sistem informasi, maka SIG juga dapat dikatakan sebagai suatu kesatuan formal yang terdiri dari berbagai sumber daya fisik dan logika yang berkenaan dengan objek-objek yang terdapat di permukaan bumi.

Sistem Informasi Geografis merupakan sejenis perangkat lunak, perangkat keras, manusia, prosedur, basis data, dan fasilitas jaringan komunikasi yang digunakan untuk memfasilitasi proses-proses pemasukan, penyimpanan, manipulasi, menampilkan, dan keluaran data/informasi geografis berikut atribut-atributnya.

II.2.1. Komponen Sistem Informasi Geografis

Menurut Eddy Prahasta (2014 : 104) Sistem Informasi Geografis merupakan sistem kompleks yang umumnya terintegrasi dengan sistem komputer lainnya di tingkat fungsional & jaringan. Jika diuraikan, Sistem Informasi Geografis terdiri dari komponen dengan berbagai karakteristiknya :

- a) **Perangkat keras.** SIG tersedia di berbagai *platform* perangkat keras; mulai dari kelas *PC desktop*, *workstations*, hingga *multi-user host*. Walaupun demikian, fungsionalitas SIG tidak terikat ketat pada karakteristik fisik perangkat kerasnya hingga keterbatasan

memori pada PC dapat diatasi. Adapun perangkat keras yang sering digunakan untuk aplikasi SIG adalah komputer (PC/CPU), *mouse*, *keyboard*, monitor (plus VGA-card grafik) yang beresolusi tinggi, *digitizer*, *printer*, *plotter*, *receiver* GPS dan *scanner*.

- b) **Perangkat lunak.** SIG merupakan sistem perangkat lunak dimana sistem basisdatanya memegang peranan kunci. Pada SIG lama, sub-sistem diimplementasikan oleh modul-modul perangkat lunak hingga tidak mengherankan jika ada perangkat SIG yang terdiri dari ratusan modul program (*.exe) yang dapat dieksekusi tersendiri.
- c) **Data & informasi geografis.** SIG dapat mengumpulkan & menyimpan data/informasi yang diperlukan baik tidak langsung (dengan meng-*import*-nya) maupun langsung dengan mendijitasi data spasialnya (*on-screen/head-ups* pada layar monitor atau cara manual dengan *digitizer*) dari peta analog dan memasukkan data atributnya dari tabel/laporan dengan menggunakan *keyboard*.
- d) **Manajemen.** Proyek SIG akan berhasil jika dikelola dengan baik & dikerjakan oleh orang yang memiliki keahlian yang tepat pada semua tingkatan.

II.3. Ragam Budaya

Menurut (kotatuaku.com 2012) Keragaman budaya atau “*cultural diversity*” adalah keniscayaan yang ada di bumi Indonesia. Keragaman budaya di Indonesia adalah sesuatu yang tidak dapat dipungkiri keberadaannya. Dalam

konteks pemahaman masyarakat majemuk, selain kebudayaan kelompok suku bangsa, masyarakat Indonesia juga terdiri dari berbagai kebudayaan daerah bersifat kewilayahan yang merupakan pertemuan dari berbagai kebudayaan kelompok sukubangsa yang ada di daerah tersebut. Dengan jumlah penduduk 200 juta orang dimana mereka tinggal tersebar di pulau-pulau di Indonesia. Mereka juga mendiami dalam wilayah dengan kondisi geografis yang bervariasi. Mulai dari pegunungan, tepian hutan, pesisir, dataran rendah, pedesaan, hingga perkotaan. Hal ini juga berkaitan dengan tingkat peradaban kelompok-kelompok suku bangsa dan masyarakat di Indonesia yang berbeda. Pertemuan-pertemuan dengan kebudayaan luar juga mempengaruhi proses asimilasi kebudayaan yang ada di Indonesia sehingga menambah ragamnya jenis kebudayaan yang ada di Indonesia. Kemudian juga berkembang dan meluasnya agama-agama besar di Indonesia turut mendukung perkembangan kebudayaan Indonesia sehingga mencerminkan kebudayaan agama tertentu. Bisa dikatakan bahwa Indonesia adalah salah satu negara dengan tingkat keanekaragaman budaya atau tingkat heterogenitasnya yang tinggi. Tidak saja keanekaragaman budaya kelompok suku bangsa namun juga keanekaragaman budaya dalam konteks peradaban, tradisional hingga ke modern, dan kewilayahan.

Dengan keanekaragaman kebudayaannya Indonesia dapat dikatakan mempunyai keunggulan dibandingkan dengan negara lainnya. Indonesia mempunyai potret kebudayaan yang lengkap dan bervariasi. Dan tak kalah pentingnya, secara sosial budaya dan politik masyarakat Indonesia mempunyai

jalinan sejarah dinamika interaksi antar kebudayaan yang dirangkai sejak dulu. Interaksi antar kebudayaan dijalin tidak hanya meliputi antar kelompok suku bangsa yang berbeda, namun juga meliputi antar peradaban yang ada di dunia. Labuhnya kapal-kapal Portugis di Banten pada abad pertengahan misalnya telah membuka diri Indonesia pada lingkup pergaulan dunia internasional pada saat itu. Hubungan antar pedagang gujarat dan pesisir jawa juga memberikan arti yang penting dalam membangun interaksi antar peradaban yang ada di Indonesia. Singgungan-singgungan peradaban ini pada dasarnya telah membangun daya elastis bangsa Indonesia dalam berinteraksi dengan perbedaan. Disisi yang lain bangsa Indonesia juga mampu menelisik dan mengembangkan budaya lokal ditengah-tengah singgungan antar peradaban itu.

II.4. Indonesia

Menurut (id.wikipedia.org) Republik Indonesia, disingkat RI atau Indonesia, adalah negara di Asia Tenggara yang dilintasi garis khatulistiwa dan berada di antara benua Asia dan Australia serta antara Samudra Pasifik dan Samudra Hindia. Indonesia adalah negara kepulauan terbesar di dunia yang terdiri dari 13.466 pulau, nama alternatif yang biasa dipakai adalah Nusantara. Dengan populasi lebih dari 237 juta jiwa pada tahun 2010, Indonesia adalah negara berpenduduk terbesar keempat di dunia dan negara yang berpenduduk Muslim terbesar di dunia, dengan lebih dari 207 juta jiwa, meskipun secara resmi bukanlah negara Islam. Bentuk pemerintahan Indonesia adalah republik,

dengan Dewan Perwakilan Rakyat, Dewan Perwakilan Daerah dan Presiden yang dipilih langsung.

Ibu kota negara ialah Jakarta. Indonesia berbatasan darat dengan Malaysia di Pulau Kalimantan, dengan Papua Nugini di Pulau Papua dan dengan Timor Leste di Pulau Timor (mantan bagian provinsi dari Indonesia). Negara tetangga lainnya adalah Singapura, Filipina, Australia, dan wilayah persatuan Kepulauan Andaman dan Nikobar di India.

Sejarah Indonesia banyak dipengaruhi oleh bangsa lainnya. Kepulauan Indonesia menjadi wilayah perdagangan penting setidaknya sejak abad ke-7, yaitu ketika Kerajaan Sriwijaya di Palembang menjalin hubungan agama dan perdagangan dengan Tiongkok dan India. Kerajaan-kerajaan Hindu dan Buddha telah tumbuh pada awal abad Masehi, diikuti para pedagang yang membawa agama Islam, serta berbagai kekuatan Eropa yang saling bertempur untuk memonopoli perdagangan rempah-rempah Maluku semasa era penjelajahan samudra. Setelah berada di bawah penjajahan Belanda, Indonesia yang saat itu bernama Hindia Belanda menyatakan kemerdekaannya di akhir Perang Dunia II. Selanjutnya Indonesia mendapat berbagai hambatan, ancaman dan tantangan dari bencana alam, korupsi, separatisme, proses demokratisasi dan periode perubahan ekonomi yang pesat.

Dari Sabang sampai Merauke, Indonesia terdiri dari berbagai suku bangsa, bahasa, dan agama. Berdasarkan rumpun bangsa (ras), Indonesia terdiri atas bangsa asli pribumi yakni Mongoloid Selatan/Austronesia dan Melanesia di mana bangsa Austronesia yang terbesar jumlahnya dan lebih banyak

mendiami Indonesia bagian barat. Berdasarkan bangsa yang lebih spesifik, suku bangsa Jawa adalah suku bangsa terbesar dengan populasi mencapai 41,7% dari seluruh penduduk Indonesia. Semboyan nasional Indonesia, "Bhinneka tunggal ika" ("Berbeda-beda tetapi tetap satu"), berarti keberagaman yang membentuk negara. Selain memiliki populasi padat dan wilayah yang luas, Indonesia memiliki wilayah alam yang mendukung tingkat keanekaragaman hayati terbesar kedua di dunia.

II.5. PHP

Menurut Adhi Prasetyo (2015 : 130-131) PHP (PHP: *Hypertext Preprocessor*) adalah bahasa script yang ditanam di sisi server. Prosesor PHP dijalankan di server (*Windows atau Linux*). Saat sebuah halaman dibuka dan mengandung kode PHP, prosesor itu akan menerjemahkan dan mengeksekusi semua perintah dalam halaman tersebut, dan kemudian menampilkan hasilnya ke browser sebagai halaman HTML biasa. Karena penerjemahan ini terjadi di server, sebuah halaman ditulis dengan PHP dapat dilihat dengan menggunakan semua jenis browser, di sistem operasi apaun.

II.5.1. Sejarah Singkat PHP

Menurut (Loka Dwiartara : 4) PHP pertama kali ditemukan pada 1995 oleh seorang Software Developer bernama Rasmus Lerdorf. Ide awal PHP adalah ketika itu Rasmus ingin mengetahui jumlah pengunjung yang membaca *resume* onlinenya. Script yang dikembangkan baru dapat melakukan 2 pekerjaan, yakni merekam informasi visitor, dan menampilkan jumlah pengunjung dari suatu website. Dan sampai sekarang kedua tugas tersebut

masih tetap populer digunakan oleh dunia web saat ini. Kemudian, dari situ banyak orang dimilisi mendiskusikan *script* buatan Rasmus Lerdorf, hingga akhirnya Rasmus mulai membuat sebuah *tool/script*, bernama Personal Home Page (PHP).

II.5.2. Keunggulan-Keunggulan PHP

Menurut Adhi Prasetyo (2015 : 133-134) Pada saat ini banyak website yang menggunakan PHP sebagai dasar pengolahan data. Beberapa keunggulan yang dimiliki PHP adalah:

1. Kesederhanaan, karena PHP memang secara khusus didesain untuk membuat aplikasi web, dan karena itu, PHP memiliki banyak sekali fungsi *built in* untuk menangani kebutuhan standar pembuatan aplikasi web. Dengan adanya fungsi-fungsi tersebut, maka tentu saja proses belajar PHP terutama dalam pengembangan aplikasi akan jauh lebih mudah karena semua sudah tersedia.
2. PHP bersifat *open source*. Karena *source code* PHP tersedia secara gratis, maka hal tersebut memungkinkan komunitas *developer* untuk selalu melakukan perbaikan, pengembangan dan menemukan *bug* dalam bahasa PHP.
3. Stabilitas dan kompatibilitas. Saat ini, PHP berjalan stabil di berbagai macam sistem operasi seperti berbagai versi UNIX (termasuk *Linux*), *Windows* dan *Mac*. PHP juga terintegrasi secara baik dengan berbagai macam web server termasuk dua yang paling populer yaitu *IIS* dan *Apache*.

4. PHP juga dilengkapi dengan berbagai macam pendukung lain seperti *support* langsung ke berbagai macam *database* yang populer, arsitektur yang dapat dikembangkan dan sebuah prosesor yang tidak hanya menggunakan *resource* minimal pada komputer anda dibandingkan kompetitornya, tetapi juga dapat menampilkan halaman web dengan cepat.

II.6. HTML

Menurut Adhi Prasetio (2015 : 98-109) HTML (*Hyper Text Markup Language*) merupakan sebuah file teks yang berisi *tag-tag markup* yang dapat memberitahukan browser bagaimana harus menampilkan sebuah halaman. *File* HTML harus memiliki ekstensi *htm* atau *html* dan *file* HTML juga dapat dibuat menggunakan editor teks seperti *Notepad* ataupun *Notepad++*. *File* HTML memiliki 2 ekstension yaitu *.htm* dan *.html*. Ekstension *.htm* ditujukan untuk operating sistem jaman dulu yang hanya *support* ekstensi 3 huruf. Sedangkan ekstensi *.html* akan lebih aman digunakan jika OS dan aplikasinya *support* karena lebih jelas menunjukkan bahwa ini adalah *file* HTML.

Dokumen HTML adalah file teks yang terdiri dari elemen HTML. Elemen HTML itu didefinisikan menggunakan apa yang disebut dengan tag HTML. Berikut poin-poin untuk pemahaman Tag HTML:

- a. Tag HTML digunakan untuk menandai (*mark-up*) elemen HTML
- b. Tah HTML berada di antara dua karakter penanda berikut *<* dan *>*
- c. Karakter penanda itu disebut dengan tanda kurung siku
- d. Tag HTML umumnya selalu berpasangan seperti ** dan **

- e. Tag pertama adalah tag pembuka, dan tag kedua adalah tag penutup
- f. Teks di antara kedua tag tersebut disebut isi elemen
- g. Tag HTML tidak bersifat *case sensitif*, memiliki arti yang sama dengan

Salah satu tag HTML yang paling penting adalah tag-tag HTML yang mendefinisikan judul <h1> sampai <h6>. <h1> mendefinisikan huruf judul yang paling besar, dan <h6> yang terkecil. Mendefinisikan paragraf <p> dan di akhiri tag penutup </p>, sama seperti judul, HTML secara otomatis akan menambahkan baris kosong sebelum dan sesudah paragraf. Mendefinisikan garis baru
, tag HTML
 akan membuatkan sebuah baris baru.

II.6.1. Atribut HTML

Menurut Adhi Prasetyo (2015 : 109) Tag HTML memiliki atribut. Atribut memungkinkan informasi tambahan pada elemen HTML. Atribut selalu memiliki pasangan nama/nilai seperti ini: nama="nilai". Atribut selalu dicantumkan pada awal tag elemen HTML. Nilai atribut seharusnya selalu diapit oleh tanda kutip. Tanda kutip ganda paling sering digunakan, tapi sebenarnya tanda kutip tunggal juga bisa digunakan.

II.7. MapInfo

Menurut Wahana Komputer (2015 : 1-2) MapInfo adalah satu software pengolah Sistem Informasi Geografis (SIG). MapInfo diminati oleh pemakai SIG karena mempunyai karakteristik yang menarik, seperti mudah digunakan, tampilan interaktif, *user friendly*, dan dapat dimodifikasi menggunakan bahasa

skrip yang dimiliki. *Software* ini memiliki kemampuan untuk mengorganisir, memanipulasi, serta menganalisa data.

Selain itu, melalui menu utama maupun ikon menu yang tersedia, akan memudahkan pengguna dalam mengoperasikan tanpa perlu menghafal perintah-perintah yang panjang. Pembentukan peta di MapInfo dapat diilustrasikan secara analog. Pada MapInfo, suatu tabel dapat digambarkan sebagai satu lembar (*sheet*) dari suatu film. Suatu komposisi peta di MapInfo merupakan gabungan dari beberapa lembar (*sheet*) film tersebut yang disusun secara bertumpuk. Istilah yang umum digunakan untuk susunan tersebut adalah “*layering*”. Setiap lembar (*sheet*) merupakan *layer* yang dapat digabungkan dan dipasangkan untuk membentuk suatu peta, sehingga dapat dilakukan analisis dari peta yang terbentuk. Satu hal yang perlu diingat adalah ketika MapInfo melakukan “*redraw*” peta, MapInfo akan melakukan *redraw* dari *layer* yang tersusun paling bawah ke *layer* di atasnya. Hal ini berlaku sebaliknya jika ingin diketahui informasi dari suatu peta.

II.7.1. Struktur Data MapInfo

Menurut Wahana Komputer (2015 : 11) Pembentukan data grafis dan atribut pada MapInfo dilakukan secara bersamaan, sehingga pada akhirnya keduanya merupakan satu kesatuan yang utuh. Perubahan pada yang satu berarti terjadi perubahan pada yang lain. Hal ini juga berlaku jika anda menghilangkan salah satunya (obyek grafis atau atributnya), berarti akan menghilangkan semuanya. Model basis data yang dibangun dengan MapInfo adalah model data relasional.

Pada pengoperasiannya, MapInfo mengorganisasikan data grafik dan data non-grafik dalam beberapa file tertentu. File tersebut dibedakan oleh ekstensi yang menyertainya. Masing-masing ekstensi menandai jenis file. Jenis-jenis file yang ada pada MapInfo dapat dibagi menjadi seperti berikut:

- a. DAT adalah file yang berisi data atribut berupa numerik, alfabetik, dan alfa numerik.
- b. ID adalah file yang berisi identifier (ID) atau kode tertentu yang bersifat unik. Identifier ini dapat dibangun secara otomatis oleh MapInfo maupun dibangun sendiri.
- c. MAP adalah file yang berisi data grafis maupun bentuk-bentuk dasar lain seperti titik, garis dan luasan.
- d. TAB adalah file teks yang menyimpan struktur tabel dan format data yang tersimpan.

II.7.2. Mengenal Data Grafis

Menurut Wahana Komputer (2015 : 12) MapInfo membagi data grafis menjadi tiga, yaitu titik, garis dan area. Objek titik hanya terdiri dari satu pasang koordinat x,y. Objek garis terdiri dari posisi x,y awal dan x,y akhir. Objek area terdiri dari beberapa pasangan x,y.

Ketiga objek tersebut dapat digunakan anatar lain sebagai berikut:

- a. Titik dapat digunakan untuk mewakili objek kota, stasiun, curah hujan, alamat, dan lain-lain.
- b. Garis dapat digunakan untuk menggambarkan jalan, sungai, jaringan listrik.

- c. Area dapat digunakan untuk mewakili batas administrasi, penggunaan lahan, kemiringan lereng, dan lain-lain.

II.8. MySQL

Menurut Loka Dwiartara (2012 : 6) MySQL adalah *Database*. *Database* sendiri merupakan suatu jalan untuk dapat menyimpan berbagai informasi dengan membaginya berdasarkan kategori-kategori tertentu. Dimana informasi-informasi tersebut saling berkaitan, satu dengan yang lainnya. MySQL bersifat RDBMS (*Relational Database Management System*) RDBMS memungkinkan seorang admin dapat menyimpan banyak informasi ke dalam tabel-tabel, dimana tabel-tabel tersebut saling berkaitan satu sama lain. Keuntungan RDBMS sendiri adalah kita dapat memecah *database* kedalam tabel-tabel yang berbeda. Setiap tabel memiliki informasi yang berkaitan dengan dengan tabel yang lainnya.

II.9. CSS

Menurut Adhi Prasetyo (2015 : 285) CSS (*Cascading Style Sheet*) adalah suatu teknologi yang digunakan untuk memperindah tampilan halaman website. Singkatnya dengan menggunakan metode css ini dapat dengan mudah mengubah secara keseluruhan sekaligus memformat ulang desain website.

II.10. UML (*Unified Modelling Language*)

Menurut Adi Nugroho (2010 : 6-7) UML (*Unified Modelling Language*) adalah ‘bahasa’ pemodelan untuk sistem atau perangkat lunak yang berparadigma ‘berorientasi objek’. Pemodelan (*modelling*) sesungguhnya

digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipahami.

Menurut Jurnal Teknik Informatika Universitas Udayana (4 : 2014) 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*. Notasi standar yang disediakan UML bisa digunakan sebagai alat komunikasi bagi para pelaku dalam proses analisis yaitu, diagram *use case*, diagram *sequence*, dan diagram *class*. Dengan menggunakan UML dapat membuat model untuk semua jenis aplikasi piranti perangkat lunak, dimana aplikasi tersebut dapat berjalan pada piranti perangkat keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka UML cocok untuk penulisan piranti lunak dalam bahasa pemrograman yang berorientasi objek.

Tabel II.1 View dan diagram dalam UML

<i>Major Area</i>	<i>View</i>	<i>Diagrams</i>	<i>Main Concepts</i>
<i>Structural</i>	<i>Static view</i>	<i>Class diagram</i>	<i>Class, association, generalization, dependency, realization, interface</i>
	<i>Use case view</i>	<i>Use case diagram</i>	<i>Use case, actor, association, extend, include, use case,</i>

			<i>generalization</i>
	<i>Implementation view</i>	<i>Component diagram</i>	<i>Component, interface, depending, realization</i>
	<i>Deployment view</i>	<i>Deployment diagram</i>	<i>Node, component, dependency, location</i>
<i>Dynamic</i>	<i>State machine view</i>	<i>Statechart diagram</i>	<i>State, event, transition, action</i>
	<i>Activity view</i>	<i>Activity diagram</i>	<i>State, activity, completion transition, fork, join</i>
	<i>Interaction view</i>	<i>Sequence diagram</i>	<i>Interaction, object, message, activation</i>
		<i>Collaboration diagram</i>	<i>Collaboration, interaction, collaboration, collaboration role, message</i>
<i>Model management</i>	<i>Model management view</i>	<i>Class diagram</i>	<i>Package, subsystem, model</i>
<i>Extensibility</i>	<i>All</i>	<i>All</i>	<i>Constraint, stereotype, tagged values</i>

(Sumber : Adi Nugroho ; 2010)

Tabel II.1 memperlihatkan hubungan antara *view-view* dan diagram-diagram yang menampilkannya, serta konsep-konsep utama yang relevan untuk masing-masing *view*. Pada peringkat paling atas, *view-view* sesungguhnya dapat dibagi menjadi 3 area utama, yaitu: Klasifikasi struktural (*structural classification*), perilaku dinamis (*dynamic behaviour*), serta pengelolaan/manajemen model (*model management*) (Perhatikan Tabel II.1 di atas).

Klasifikasi struktural (*structural classification*) mendeskripsikan segala sesuatu dalam sistem/perangkat lunak dan relasinya terhadap sesuatu yang lainnya. Pengklasifikasi (*classifier*) mencakup di dalamnya kelas-kelas (*class*), *use case-use case*, komponen-komponen (*component*), dan simpul-simpul (*node*). Pengklasifikasi (*classifier*) menyediakan dasar dimana perilaku dinamis sistem/perangkat lunak dikembangkan. *View* klasifikasi mencakup di dalamnya *view* statis (*static view*), *view use case*, serta *view* implementasi. Perilaku dinamis (*dynamic view*) mendeskripsikan perilaku sistem selama berjalannya waktu. Perilaku (*behaviour*) sendiri sesungguhnya dapat dideskripsikan sebagai deretan perubahan sistem/perangkat lunak dari suatu *view* statis. *View* perilaku dinamis mencakup di dalamnya *state machine view*, *activity view*, dan *interaction view*. Sementara itu, pengelolaan model mendeskripsikan organisasi model-model itu sendiri menjadi unit-unit yang bersifat hierarki. Paket (*package*) merupakan unit pengorganisasi generik untuk model-model. Paket-paket khusus (*package*) mencakup di dalamnya model-model dan subsistem-subsistem. *View* pengelolaan model (*model management view*)

bersifat lintas-view dan mengorganisasinya sedemikian rupa untuk memudahkan pekerjaan pengembangan sistem/perangkat lunak dan memudahkan kendali konfigurasi.

II.11. Use Case Diagram

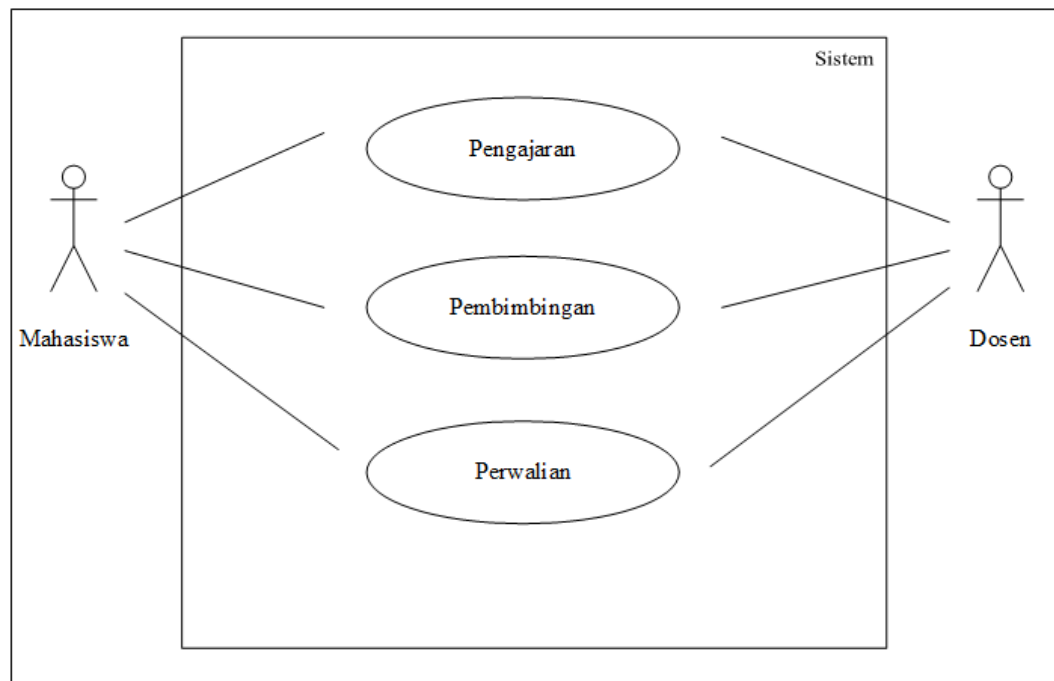
Menurut Adi Nugroho (2010 : 34-35) *Use Case* sesungguhnya merupakan unit koheren dari fungsionalitas sistem/perangkat lunak yang tampak dari luar dan diekspresikan sebagai urutan pesan-pesan yang dipertukarkan unit-unit sistem dengan satu atau lebih *actor* yang ada di luar sistem.

Tabel II.2 Komponen Use Case Diagram

No.	Nama Komponen	Keterangan	Simbol
1.	<i>Actor</i>	<i>Actor</i> juga dapat berkomunikasi dengan <i>object</i> , maka <i>actor</i> juga dapat diurutkan sebagai kolom. Simbol <i>Actor</i> sama dengan simbol pada <i>Actor Use Case Diagram</i> .	
2.	<i>Association</i>	<i>Association</i> atau Asosiasi digunakan untuk menghubungkan <i>actor</i> dengan <i>use case</i> . Asosiasi digambarkan dengan sebuah garis yang menghubungkan antara <i>actor</i> dengan <i>use case</i> .	
3.	<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.	
4.	<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i> .	

(Sumber : Grady Booch ; 2013)

Kegunaan dari *use case* adalah untuk mendaftarkan *actor-actor* dan *use case-use case* dan memperlihatkan *actor-actor* mana yang berpartisipasi dalam masing-masing *use case*.



Gambar II.1 Use Case

(Sumber : Adi Nugroho ; 2010 : 34)

Gambar II.1 memperlihatkan contoh penggambaran diagram *use case*. Ikon-ikon berbentuk orang adalah *actor-actor*-nya sementara elips-elips yang ada menggambarkan fungsionalitas-fungsionalitas sistem. Sebuah (bukan seorang!) *actor* merupakan idealisasi dari orang yang ada di luar sistem, idealisasi proses-proses yang berinteraksi dengan sistem, atau idealisasi sesuatu yang berinteraksi dengan sistem, subsistem, atau kelas, pada sistem/perangkat lunak yang sedang kita kembangkan. Sebuah *actor* pada dasarnya

menggambarkan interaksi pengguna-pengguna sistem dengan sistem/perangkat lunak yang sedang kita kembangkan. Saat sistem/perangkat lunak dijalankan, suatu pengguna secara fisik mungkin diikat menjadi *actor* yang ada dalam sistem dan selanjutnya mempresentasikan *instance* majemuk dari definisi *actor* yang sama.

Use case sesungguhnya merupakan unit koheren dari fungsionalitas sistem/perangkat lunak yang tampak dari luar dan diekspresikan sebagai urutan pesan-pesan yang dipertukarkan unit-unit sistem dengan satu atau lebih *actor* yang ada di luar sistem. Kegunaan *use case* sesungguhnya adalah untuk mendefinisikan suatu bagian perilaku sistem yang bersifat koheren tanpa perlu menyingkapkan struktur internal sistem/perangkat lunak yang sedang dikembangkan.

II.12. Sequence Diagram

Menurut Adi Nugroho (2010 : 42-43) *Sequence Diagram* memperlihatkan interaksi sebagai diagram dua matra (dimensi). Matra vertikal adalah sumbu waktu; waktu bertambah dari atas ke bawah. Matra horizontal memperlihatkan peran pengklasifikasi yang merepresentasikan objek-objek mandiri yang terlibat dalam kolaborasi. Masing-masing peran pengklasifikasi direpresentasikan sebagai kolom-kolom vertikal dalam *sequence diagram* sering disebut garis waktu (*lifeline*). Selama objek ada, peran digambarkan menggunakan garis tegas. Selama aktivasi prosedur pada objek aktif, garis waktu digambarkan sebagai garis tanda. Pesan-pesan digambarkan sebagai suatu tanda panah dari garis waktu suatu objek ke garis waktu objek lainnya.

Panah-panah yang menggambarkan aliran pesan antarperan pengklasifikasi digambarkan dalam urutan waktu kejadiannya dari atas ke bawah.

II.13. Class Diagram

Menurut Adi Nugroho (2010 : 17) Kelas sesungguhnya merepresentasikan suatu konsep diskret di dalam aplikasi yang dimodelkan sesuatu yang bersifat fisik (misalnya mobil, pesawat terbang, dan sebagainya), sesuatu yang bersifat bisnis (misalnya pesanan), sesuatu yang bersifat logika (misalnya tombol-tombol, ikon-ikon, dan sebagainya), sesuatu yang merupakan konsep yang dikenali dalam terminologi ilmu komputer (misalnya tabel *hash* atau berbagai metode pengurutan dan pencarian [*sorting* dan *searching*]), atau sesuatu yang bersifat perilaku (*behaviour*) (misalnya pekerjaan tertentu).

II.14. Collaboration Diagram

Menurut Adi Nugroho (2010 : 44) Collaboration diagram pada dasarnya merupakan diagram kelas yang memuat peran-peran pengklasifikasi dan peran-peran asosiasi, alih-alih hanya menampilkan pengklasifikasi-pengklasifikasi serta asosiasi-asosiasi. Peran pengklasifikasi dan peran asosiasi mendeskripsikan konfigurasi objek-objek dan tautan-tautan yang mungkin terjadi saat suatu *instance* kolaborasi dieksekusi. Saat kolaborasi dibentuk, objek-objek yang terikat ke peran-peran pengklasifikasi dan tautan-tautan diikat ke peran asosiasi.

II.15. Activity Diagram

Menurut (Adi Nugroho, 2010 : 62) Diagram aktivitas (*activity diagram*) sesungguhnya merupakan bentuk khusus dari *state machine* yang bertujuan

memodelkan komputasi-komputasi dan aliran-aliran kerja yang terjadi dalam sistem/perangkat lunak yang sedang dikembangkan. Biasanya, suatu diagram aktivitas mengasumsikan komputasi-komputasi dilaksanakan tanpa adanya interupsi-interupsi eksternal berbasis *event* terjadi padanya.

II.16. Basis Data

Menurut Kusrini (2011 : 2) Basis data adalah kumpulan data yang saling berelasi. Data sendiri merupakan fakta mengenai obyek, orang, dan lain-lain. Data dinyatakan dengan nilai (angka, deretan karakter, atau simbol). Basis data dapat didefinisikan dalam berbagai sudut pandang seperti berikut :

- a. Himpunan kelompok data yang saling berhubungan yang diorganisasi sedemikian rupa sehingga kelak dapat dimanfaatkan dengan cepat dan mudah.
- b. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa tanpa pengulangan (*redundancy*) yang tidak perlu, untuk memenuhi kebutuhan.
- c. Kumpulan file/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.

II.17. Normalisasi

Menurut Kusrini (2011 : 39-43) Normalisasi merupakan cara pendekatan dalam membangun desain logika basis data relasional yang tidak secara langsung berkaitan dengan model data, tetapi dengan menerapkan

sejumlah aturan dan kriteria standar untuk menghasilkan struktur tabel yang normal. Pada dasarnya desain logika basis data relasional dapat menggunakan prinsip normalisasi maupun transformasi dari E-R ke bentuk fisik.

Dalam perpektif normalisasi sebuah database dikatakan baik jika setiap tabel yang membentuk basis data sudah berada dalam keadaan normal. Suatu tabel dikatakan normal jika :

- a. Jika ada dekomposisi/penguraian tabel, maka dekomposisinya dijamin aman (*lossless-join decomposition*)
- b. Terpeliharanya ketergantungan functional pada saat perubahan data (*dependency preservation*)
- c. Tidak melanggar *Boyce Code Normal Form* (BCNF), jika tidak bisa minimal tidak melanggar bentuk normalisasi ketiga

Yang dimaksud dengan ketergantungan fungsional (*functional dependency*) adalah :

Diberikan sebuah tabel/relasi T, atribut B dari T bergantung secara fungsi pada atribut A dari T jika dan hanya jika setiap nilai B dari T punya hubungan dengan tepat satu nilai A dalam T (dalam setiap satu waktu).

Beberapa kondisi yang diujikan pada proses normalisasi :

- a. Menambah data/*insert*
- b. Mengedit/*update*
- c. Menghapus/*delete*
- d. Membaca/*retrieve*

II.17.1. Bentuk-bentuk Normalisasi

a. Bentuk tidak normal

Bentuk ini merupakan kumpulan data yang akan direkam, tidak ada keharusan mengikuti format tertentu, dapat saja tidak lengkap dan terduplikasi. Data dikumpulkan apa adanya sesuai keadaannya.

b. Bentuk normal tahap pertama (1NF)

Sebuah *table* disebut 1NF jika memenuhi syarat-syarat sebagai berikut:

- 1. Tidak ada baris yang duplikasi dalam tabel tersebut
- 2. Masing-masing cell bernilai tunggal

Catatan : permintaan yang menyatakan tidak ada baris yang duplikasi dalam sebuah tabel berarti tabel tersebut memiliki sebuah kunci, meskipun kunci tersebut merupakan kombinasi dari semua kolom.

Contoh:

- 1. Tabel kuliah (kode_kul, nama_kul, sks, semester, nama_dos)
- 2. Tabel jadwal (kode_kul, waktu, ruang)

c. Bentuk normal tahap kedua (2NF)

Bentuk normal kedua (2NF) terpenuhi jika pada sebuah tabel semua atribut yang tidak termasuk dalam *primary key* memiliki ketergantungan fungsional pada *primary key* secara utuh. Sebuah tabel dikatakan tidak memenuhi 2NF jika ketergantungannya hanya bersifat parsial (hanya tergantung pada sebagian dari *primary key*).

Contoh:

Misal tabel nilai terdiri dari atribut kode_kul, nim dan nilai. Jika pada tabel nilai, misalnya kita tambahkan sebuah atribut yang bersifat redundan, yaitu nama_mhs, maka tabel nilai ini dianggap melanggar 2NF. *Primary key* pada tabel nilai adalah (kode_kul, nim). Penambahan atribut baru (nama_mhs) akan menyebabkan adanya ketergantungan fungsional yang baru yaitu $\text{nim} > \text{nama_mhs}$. Karena atribut nama_mhs ini hanya memiliki ketergantungan parsial pada *primary key* secara utuh (hanya tergantung pada nim, padahal nim hanya bagian dari *primary key*). Bentuk normal kedua ini dianggap belum memadai karena meninjau sifat ketergantungan atribut terhadap *primary key*.

d. Bentuk normal tahap ketiga (3NF)

Sebuah tabel dikatakan memenuhi bentuk normal ketiga (3NF), jika untuk setiap ketergantungan fungsional dengan relasi $X > A$, dimana A mewakili semua atribut unggal di dalam tabel yang tidak ada di dalam X, maka :

1. X haruslah superkey pada tabel tersebut
2. Atau A merupakan bagian dari *primary key* pada tabel tersebut

Misalkan pada tabel mahasiswa, atribut `alamat_mhs` dipecah kedalam `alamat_jalan`, `alamat_kota` dan `kode_pos`. Bentuk ini tidak memenuhi 3NF, karena terdapat ketergantungan fungsional baru yang muncul pada tabel tersebut, yaitu :

$$\text{alamat_jalan nama_kota} > \text{kode_pos}$$

dalam hal ini (`alamat_jalan`, `nama_kota`) bukan superkey sementara `kode_pos` juga bukan bagian dari primery key pada tabel mahasiswa. Jika tabel mahasiswa didekomposisi menjadi tabel mahasiswa dan tabel alamat, maka telah memenuhi 3NF. Hal itu dapat dibuktikan dengan memeriksa dua ketergantungan fungsional pada tabel alamat tersebut, yaitu :

$$\text{alamat_jalan nama_kota} > \text{kode_pos}$$

$$\text{kode_pos} > \text{nama_kota}$$

ketergantungan fungsional yang pertama tidak melanggar 3NF, karena (`alamat_jalan`, `nama_kota`) merupakan superkey (sekalius sebagai *primary key*) dari tabel alamat tersebut. Demikian juga dengan ketergantungan fungsional yang kedua meskipun (`kode_pos`) bukan merupakan superkey, tetapi `nama_kota` merupakan bagian dari *primary key* dari tabel alamat. Karena telah memenuhi 3NF, maka tabel tersebut tidak perlu didekomposisi lagi.

e. Bentuk normal tahap keempat dan kelima

Penerapan aturan normalisasi sampai bentuk normal ketiga sudah memadai untuk menghasilkan tabel berkualitas baik. Namun demikian, terdapat pula bentuk normal keempat (4NF) dan kelima (5NF). Bentuk 4NF berkaitan dengan sifat ketergantungan banyak nilai (*multivalued dependency*) pada suatu tabel yang merupakan pengembangan dari ketergantungan fungsional. Adapun bentuk 5NF merupakan nama lain dari *project join normal form* (PJNF).

f. Boyce Code Normal Form (BCNF)

1. Memenuhi 1NF
2. Relasi harus bergantung fungsi pada atribut superkey