



BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Konsep SCADA System

2.1.1.1 Definisi SCADA System

Dunia industri terus berkembang dengan sistem-sistem yang baru dalam bidang manufaktur, khususnya sistem kontrol. Saat ini banyak sekali ditawarkan suatu metode kontrol yang efektif dan mudah untuk di implementasikan, salah satunya *SCADA System*. (Susiono,dkk , 2006:37)

SCADA adalah suatu sistem yang dirancang untuk melakukan pengawasan dan pengendalian suatu proses secara terintegrasi, yang mencakup fungsi *monitoring* dan pengumpulan data.

a. *Supervisory Control*

Mencakup kemampuan untuk melakukan perintah *Start / Stop*, mengubah parameter suatu proses serta mengubah set point alarm.

b. *Data Acquisition*

Merupakan suatu metode yang digunakan untuk mengakses dan mengontrol informasi atau data dari proses atau peralatan yang diawasi dan di kontrol. Data tersebut kemudian dikirimkan ke lokasi yang berbeda melalui suatu media komunikasi. (Enggari, 2018:61)

Dari pengertian di atas dapat disimpulkan bahwa, *SCADA system* adalah suatu kesatuan dari beberapa peralatan yang saling berkomunikasi untuk menjalankan fungsi pengawasan, pengontrolan, dan pengumpulan data dari suatu proses. (Pujotomo, 2016:55)

2.1.1.2 Fungsi SCADA System

SCADA merupakan suatu sistem untuk pengendalian dan pemantauan jarak jauh yang memungkinkan pengguna/operator untuk melakukan :

1. *Monitoring* (pengawasan)
2. *Controlling* (pengendalian)



3. *Data Acquisition* (pengambilan dan perekaman data)

Ketiga fungsi di atas dapat dipenuhi dengan mewujudkannya dalam bentuk hardware maupun software. Dengan dukungan *SCADA system* (proses industri bisa diawasi dan dikendalikan dari jauh, sehingga bisa menghemat biaya, waktu dan tenaga), akan semakin memberikan gambaran tentang kondisi sebenarnya yang ada dalam dunia industri. (Susiono,dkk , 2006:37-38).

2.1.1.3 Prinsip *SCADA System*

1. Memantau dan mengontrol semua peralatan yang terdapat pada suatu sistem dari jarak jauh.
2. *SCADA* bekerja mengumpulkan informasi, kemudian mentransfernya ke sentral dengan membawa data-data dan sinyal kontrol (status) yang kemudian diperagakan pada sejumlah layar operator.

SCADA System dapat difungsikan dari pusat kontrol melalui komputer utama (*main computer*). Dari tampilan pada layar komputer utama inilah dispatcher dapat melakukan pengawasan dan pengendalian. (Pujotomo, 2016:55-56)

2.1.1.4 Komponen *SCADA System*

Dalam suatu *SCADA System* biasanya terdiri dari komponen-komponen sebagai berikut :

a. Unit terminal jarak jauh (*Remote Station*)

Merupakan suatu instrumen yang langsung dihubungkan dengan alat yang diawasi dan dikontrol. *Remote Station* memberikan kemampuan untu mengontrol dan mengambil data dari peralatan yang dimonitor kemudian mengirimkannya ke pusat kendali *SCADA*. *Remote Station* dapat berupa *Programable Logic Control* (PLC) maupun *Remote Terminal Unit* (RTU).

Perbedaan antara PLC dan RTU adalah dari segi media komunikasi dan dalam memprogram alat tersebut. PLC tidak di menyediakan komunikasi data dengan wireless biasanya digunakan pada plant/pabrik kecil. Sedangkan RTU menyediakan komunikasi data dengan wireless tetapi dalam pemrogramannya tidak sebaik PLC.



b. *Human Machine Interface* (HMI)

Merupakan suatu interface antara operator dengan *controller*. HMI berupa panel kontrol yang didalamnya terdapat numerik keypad dan layar LCD yang menampilkan pesan. Numerik keypad digunakan untuk memasukan data kedalam sistem kemudian LCD menampilkan pesannya.

c. *Central Monitoring Station* (CMS)

Ini merupakan pusat dari *SCADA system*, merupakan suatu tempat pengumpulan data yang diambil dari remote station dan melakukan pengontrolan jika diperlukan. CMS terdiri dari sebuah komputer atau lebih menyediakan informasi atau data dari sitem *SCADA system*. (Enggari, 2018:62)

2.1.2 Konsep Produksi Air

2.1.2.1 Definisi Produksi Air

Produksi adalah aktivitas untuk menciptakan atau menambah *utility* suatu barang atau jasa agar dapat memenuhi kebutuhan manusia. Kegiatan produksi dapat dilaksanakan apabila tersedia faktor *input* yang dibutuhkan untuk dapat menghasilkan *output* yang diinginkan. (Purba dalam Linsen, 2017:611)

Sedangkan, Air merupakan salah satu kebutuhan yang menjamin kelangsungan hidup setiap makhluk hidup. Kebutuhan terhadap air untuk keperluan sehari-hari di lingkungan rumah tangga ternyata berbeda untuk tiap tempat. (Lupiyadi dalam Dinah dan Surya, 2014:30).

Dalam kehidupan manusia, air memiliki peranan yang sangat penting. Dengan semakin cepatnya tingkat perkembangan pemukiman penduduk suatu kecamatan maupun desa, semakin banyak pula air yang dibutuhkan. Bagi rumah tangga, air bersih dipergunakan untuk minum, mandi, mencuci serta membersihkan rumah dan perabotan lainnya. Disamping keperluan rumah tangga, air juga dipergunakan sebagai penunjang lanjutnya pembangunan seperti keperluan pertanian, pembangkitan tenaga listrik, bahan baku dan bahan pembantu bagi industri-industri dan lain sebagainya. Begitu vitalnya keberadaan air, sehingga manusia selalu berusaha untuk mendapatkannya dalam jumlah yang cukup banyak. (Mailindra dan Azwar, 2018:443)



Persediaan produksi dalam perusahaan sangat penting maka diperlukan perencanaan yang baik dalam pengendalian bahan baku yang optimal. Hal ini dikarenakan pihak perusahaan tidak mempertimbangkan permintaan dan kapasitas sumberdaya untuk produksi yang dimiliki (kapasitas mesin, tenaga kerja, teknologi yang dimiliki, dan informasi). (Amri dan Efrida, 2012:11)

2.1.2.2 Proses Produksi Air

Instalasi Pengolahan Air (IPA) merupakan suatu bangunan yang digunakan untuk mengolah air bersih. Proses pengolahan air bersih melalui beberapa tahap yaitu :

1. Intake

Intake merupakan awal dari proses sebuah IPA. Dari intake ini air baku akan diteruskan ketempat penampungan sementara untuk proses selanjutnya. Volume air yang berada dalam bak penampungan sementara harus dijaga agar tetap stabil, sehingga proses pengolahan air dapat berlansung terus menerus.

2. Prasedimentasi

Proses ini dilakukan bertujuan untuk mengendapkan partikel kasar atau partikel diskret yaitu partikel yang tidak mengalami perubahan bentuk selama mengendap di dalam air.

3. Koagulasi

Air yang berada di bak penampung sementara dipompakan ke bak koagulan untuk diberi koagulan. Pada bak koagulan diharapkan partikel-partikel koloid dapat menjadi partikel flok yang lebih besar sehingga mudah mengendap. Selain ditambahkan koagulan juga dilakukan pengadukan dengan cepat menggunakan mixer, dengan tujuan untuk mencampur antara koagulan dengan koloid.

4. Flokulasi dan Sedimentasi

Dari bak koagulasi air dilairkan keunit flokulasi dan sedimentasi secara gravitasi. Sistem sedimentasi yang digunakan adalah sistem cone dengan aliran vertikal (*upflow*) yang terdiri dari 2 bak yang disusun secara seri. Proses sedimentasi dilakukan setelah proses upflow flokulasi, yaitu setelah partikel-partikel yang lebih kecil bergabung atau tersedimentasi pada partikel-partikel



yang lebih besar pada sludge blanket. Dan selanjutnya air dialirkan secara gravitasi ke tangki filter untuk proses filtrasi

5. Filtrasi

Filtrasi yang dipakai adalah rapid sand filter atau saringan pasir cepat. Proses filtrasi dimaksudkan untuk menyisihkan partikel koloid yang tidak dapat disisihkan pada proses sebelumnya dan juga untuk mengurangi jumlah bakteri organisme lain.

6. Desinfeksi

Proses ini bertujuan membunuh bakteri patogen yang ada dalam air. Desinfektan air dapat dilakukan dengan berbagai cara antara lain dengan pemanasan, penyinaran sinar UV, dan dengan senyawa kimia.

7. *Reservoir*

Hasil dari proses pengolahan ditampung di *reservoir* sebelum didistribusikan ke rumah pelanggan. (Enggari, 2018:60-61)

2.1.2.3 Alur Kerja Proses Produksi Air Dengan Pemanfaatan SCADA System di PDAM Tirta Musi Palembang

Berikut alur kerja dari proses produksi air dengan pemanfaatan *SCADA System* di PDAM Tirta Musi Palembang :

1. Pegawai bagian produksi memantau *SCADA System* dengan menggunakan computer utama di ruangan.
2. Pegawai mencatat data produksi air di catatan harian produksi air.
3. Pegawai membuat laporan bulanan dari catatan harian produksi air.
4. Pegawai memberikan laporan kepada manager untuk kemudian diarsip.

2.1.3 Konsep Monitoring

2.1.3.1 Definisi Monitoring

Ada banyak definisi tentang *monitoring* (pemantauan). Kendati definisi tersebut disajikan dengan bahasa berbeda, namun memiliki pengertian yang sama. Menurut Prijambodo (2014:9-10), *Monitoring* adalah kegiatan untuk mengamati perkembangan pelaksanaan program atau proyek. Dengan *Monitoring*, dapat



diketahui program atau proyek berjalan, sesuai dan/atau kurang sesuai dengan rencana.

Menurut Peraturan Pemerintah Nomor 36 Tahun 2006, disebutkan bahwa *monitoring* merupakan suatu kegiatan mengamati secara saksama suatu keadaan atau kondisi, termasuk juga perilaku atau kegiatan tertentu, dengan tujuan agar semua data masukan atau informasi yang diperoleh dari hasil pengamatan tersebut dapat menjadi landasan dalam mengambil keputusan tindakan selanjutnya yang diperlukan.

Mulyono dan Yumari (2017:9) menjelaskan bahwa *Monitoring* merupakan suatu kegiatan mengamati secara seksama suatu keadaan atau kondisi, termasuk juga perilaku atau kegiatan tertentu, dengan tujuan agar semua data masukan atau informasi yang diperoleh dari hasil pengamatan tersebut dapat menjadi landasan dalam mengambil keputusan tindakan selanjutnya yang diperlukan.

2.1.3.2 Jenis *Monitoring*

Menurut Prijambodo (2014:15), Secara garis besar ada 2 (dua) jenis *Monitoring*, yaitu :

1. *Monitoring* Aspek Proses

Monitoring aspek proses merupakan pekerjaan untuk mengamati apakah proses kerja yang direncanakan dapat berjalan secara baik atau tidak. *Monitoring* aspek proses di konsentrasikan pada pengamatan, berfungsi atau kurang berfungsinya pelaksanaan organisasi, manajemen prosedur dan aspek-aspek lain yang telah ditetapkan.

2. *Monitoring* Aspek Hasil

Monitoring aspek hasil difokuskan untuk mengamati perkembangan pencapaian *output*. Suatu perencanaan yang baik, sejak awal harus menetapkan target/*output* serta tahapan pencapaian.



2.1.3.3 Fungsi *Monitoring*

Pada saat menyusun rencana, semua perangkat keras, maupun perangkat lunak, serta berbagai faktor eksternal telah diperhitungkan dengan cermat. Perencanaan yang baik diharapkan berlanjut pada pelaksanaan yang baik pula. Tetapi setelah pelaksanaan dapat saja terjadi hal-hal yang tidak diharapkan. Rencana yang disusun tidak sepenuhnya berjalan sesuai kondisi yang ada. Kesiapan dan perhitungan-perhitungan seperti diatas, dalam pelaksanaan tidak selalu berjalan mulus, Sebagai akibatnya, *output* tidak tercapai sesuai rencana.

Melalui *monitoring* diperoleh data perkembangan pencapaian *output*. Jika data hasil *monitoring* ini menunjukkan ada penyimpangan, maka menjadi umpan balik (*feed back*) untuk perbaikan (koreksi). Umpan balik (*feed back*) ini dapat berupa perbaikan proses pekerjaan, sampai perbaikan (koreksi) rencana yang ada. Dengan demikian *monitoring* sekaligus berfungsi sebagai sarana pengendalian pelaksanaan, agar program atau proyek tetap berjalan sesuai yang direncanakan. . (Priambodo, 2014:12-13)

2.1.4 Metode Pengembangan Sistem

Rosa, Shalahudin (2013:26) Metode Pengembangan Sistem atau *System Development Life Cycle (SDLC)* adalah proses mengembangkan atau mengubah suatu perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan untuk mengembangkan sistem-sistem perangkat lunak sebelumnya.

Rosa, Shalahudin (2013:28-30) Metode SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life circle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian dan tahap pendukung (*support*).

1. Analisis kebutuhan perangkat lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk mespesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh user. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.



2. Desain

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

3. Pembuatan kode program

Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

4. Pengujian

Pengujian fokus pada perangkat lunak secara dari segi logik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

5. Pendukung (*support*) atau pemeliharaan (*maintenance*)

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke user. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

2.2 Teori Khusus

2.2.1 *Unified Modeling Language (UML)*

Rosa dan Shalahuddin (2013:133), menjelaskan tentang pengertian *Unified Modeling Language* sebagai berikut : “*Unified Modeling Language (UML)* adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek.”



Sedangkan menurut Braun, dkk. (Haviluddin, 2011:1) “*Unified Modelling Language* (UML) adalah suatu alat untuk memvisualisasikan dan mendokumentasikan hasil analisa dan desain yang berisi sintak dalam memodelkan sistem secara visual”.

Disini penulis memilih *UML* karena menyediakan serangkaian gambar dan diagram yang sangat baik. Beberapa diagram memfokuskan diri pada ketangguhan teori *object-oriented* dan sebagian lagi memfokuskan pada detail rancangan dan konstruksi. Semua dimaksudkan sebagai sarana komunikasi antar *team programmer* maupun dengan pengguna.



Gambar 2.1 Tampilan Logo UML
(Sumber: mahesa.id)

2.2.2 Jenis-Jenis UML

2.2.2.1 Use Case Diagram

Use case adalah rangkaian atau uraian sekelompok yang saling terikat dan membentuk system secara teratur yang dilakukan atau diawasi oleh sebuah aktor. *Use case* digunakan untuk membentuk tingkah laku benda dalam sebuah model serta direalisasikan oleh sebuah kolaborasi. (Tohari , 2014:47-49)

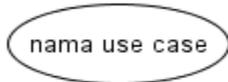
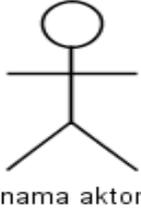
Langkah- langkah yang dibutuhkan untuk menyusun diagram use case :

1. Menidentifikasi pelaku bisnis.
2. Mengidentifikasi use case persyaratan bisnis.
3. Membuat diagram model use case.
4. Mendokumentasikan naratif use case persyaratan bisnis.



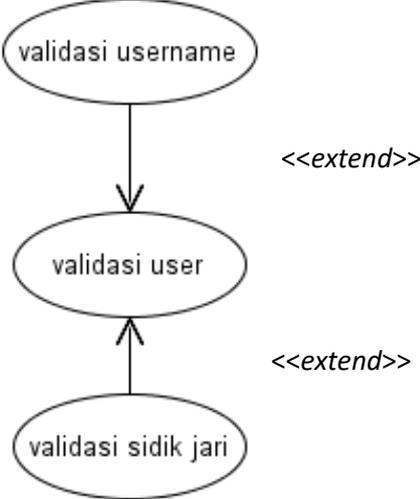
Adapun simbol-simbol yang digunakan dalam *use case diagram* adalah sebagai berikut :

Tabel 2.1 Simbol-simbol *Use case Diagram*

No	Simbol	Deskripsi
1.	<p><i>Use case</i></p> 	<p>fungsi yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal-awal frase nama <i>use case</i>.</p>
2.	<p>aktor / <i>actor</i></p> 	<p>orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.</p>
3.	<p>asosiasi / <i>association</i></p> 	<p>komunikasi antar aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.</p>

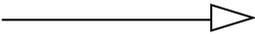
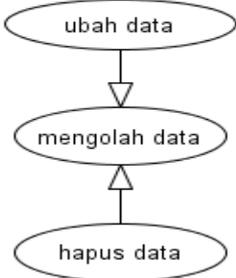
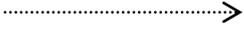
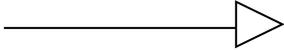
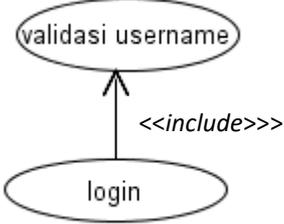


Lanjutan Tabel 2.1 Simbol-simbol Use case Diagram

4.	ekstensi / <i>extend</i> <<extend>>>	<p>relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang di tambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, misalnya</p>  <pre> graph TD A([validasi sidik jari]) -- "<<extend>>" --> B([validasi user]) C([validasi username]) -- "<<extend>>" --> B </pre> <p>biasanya <i>use case</i> yang menjadi <i>extend</i>-nya merupakan jenis yang sama dengan <i>use case</i> yang menjadi induknya.</p>
----	--	--

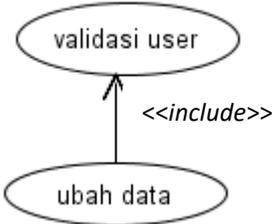


Lanjutan Tabel 2.1 Simbol-simbol Use case Diagram

No	Simbol	Deskripsi
5.	Generalisasi/ <i>generalization</i> 	<p>hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya,</p> <div style="text-align: center;">  </div> <p>misalnya: arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum).</p>
6.	menggunakan/ <i>include / uses</i>  << <i>include</i> >>  << <i>uses</i> >>	<p>relasi tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini</p> <p>ada dua sudut pandang yang cukup besar mengenai include di <i>use case</i>:</p> <ul style="list-style-type: none"> • <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu di panggil saat <i>use case</i> tambahan dijalankan, misalnya pada kasus berikut: <div style="text-align: center;">  </div>



Lanjutan Tabel 2.1 Simbol-simbol Use case Diagram

No	Simbol	Deskripsi
		<ul style="list-style-type: none"> • <i>Include</i> berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang di tambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan, misal pada kasus berikut:  <pre> graph BT UC1(ubah data) -- "<<include>>" --> UC2(validasi user) </pre> <p>kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.</p>

(Sumber: Rosa dan Shalahuddin (2013:156-158))

2.2.2.2 Activity Diagram

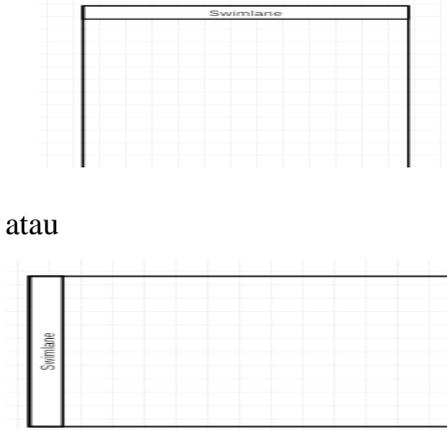
Activity diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. (Rosa dan Shalahuddin, 2013:161)

Adapun simbol-simbol yang digunakan dalam *activity diagram* adalah sebagai berikut :

Tabel 2.2 Simbol-simbol Activity Diagram

No	Simbol	Deskripsi
1.		Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal

Lanjutan Tabel 2.2 Simbol-simbol *Activity Diagram*

No	Simbol	Deskripsi
2.	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
3.	Percabangan / <i>decision</i> 	Asosiasi percabangan di mana jika ada pilihan aktivitas lebih dari satu
4.	Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
5.	Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
6.	Swimlane  atau	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

(Sumber: Rosa dan Shalahuddin (2013:162-163))

2.2.2.3 Class Diagram

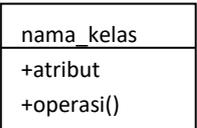
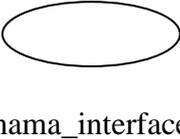
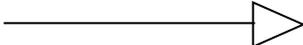
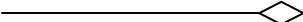
Class Diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Diagram kelas dibuat agar pembuat program atau *programmer* membuat kelas-kelas sesuai rancangan di



dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron. (Rosa dan Shalahuddin, 2013:141-142)

Adapun simbol-simbol yang digunakan dalam *class* diagram adalah sebagai berikut :

Tabel 2.3 Simbol-simbol *Class* Diagram

No	Simbol	Deskripsi
1.	kelas 	Kelas pada struktur sistem
2	antarmuka / interface 	Sama dengan konsep interface dalam pemrograman berorientasi objek
3.	asosiasi / association 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai <i>multiplicity</i>
4.	asosiasi berarah / <i>directed association</i> 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
5.	generalisasi 	Relasi antarkelas dengan makna generalisasi – spesialisasi (umum khusus)
6.	kebergantungan / <i>dependency</i> 	Relasi antarkelas dengan makna kebergantungan antar kelas
7.	agregasi / <i>aggregation</i> 	Relasi antarkelas dengan makna semua-bagian (<i>whole-part</i>)

(Sumber: Rosa dan Shalahuddin (2013:146-147))

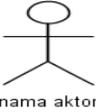
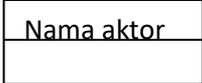


2.2.2.4 Sequence Diagram

“Diagram sekuen menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah beserta metode-metode yang dimiliki kelas yang diinstansikan menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada use case” (Rosa dan Shalahuddin, 2013:165).

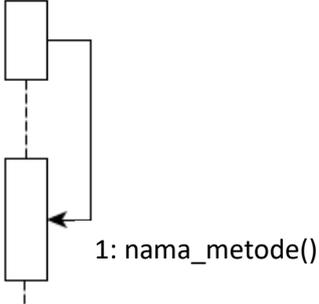
Berikut simbol-simbol pada Sequence Diagram :

Tabel 2.4 Simbol-simbol pada Sequence Diagram

No	Simbol	Deskripsi
1.	<p>Actor</p>  <p>atau</p>  <p>tanpa waktu aktif</p>	<p>orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan 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</p>
2.	<p>Garis hidup / <i>lifeline</i></p> 	<p>menyatakan kehidupan suatu objek.</p>
3.	<p>Objek</p> 	<p>menyatakan objek yang berinteraksi pesan</p>

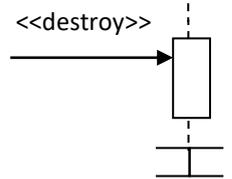


Lanjutan Tabel 2.4 Simbol-simbol pada Sequence Diagram

No	Simbol	Deskripsi
4.	Waktu aktif 	menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya.
5.	Pesan tipe create <code><<create>></code> 	menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat
6.	Pesan tipe call <code>1: nama metode()</code> 	menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri,  arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi
7.	Pesan tipe send <code>1: masukan</code> 	menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim



Lanjutan Tabel 2.4 Simbol-simbol pada Sequence Diagram

No	Simbol	Deskripsi
8.	Pesan tipe return 1: keluaran 	menyatakan bahwa suatu objek yang telah menjalankan suatu operasi menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian
9.	Pesan tipe destroy 	menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i>

(Sumber: Rosa dan Shalahuddin (2013:165-167))

2.3 Teori Program

2.3.1 Pengertian HTML

Menurut Pamungkas (2017:1-2), “HTML (Hypertext Markup Language) merupakan bahasa pemrograman web berupa suatu format data yang digunakan untuk membuat dokumen hypertext yang dapat dibaca, diinterpretasikan dari satu platform computer keplatform computer lainnya tanpa perlu melakukan suatu perubahan apapun.”

Sedangkan menurut Larry (2012:4), “Pada awalnya HTML dikembangkan sebagai subset SGML (Standard Generalized Markup Language) karena HTML didedikasikan untuk ditransmisikan melalui media internet, maka HTML relative lebih sederhana daripada SGML yang lebih ditekankan pada format dokumen yang berorientasi pada aplikasi.”

2.3.2 Pengertian Basis Data

Rosa dan Shalahuddin (2013:43), “Basis data adalah sistem komputerisasi yang tujuan utamanya adalah memelihara data yang sudah ada yang diolah atau informasi dan membuat informasi tersedia saat dibutuhkan.”



Winarno dan Utomo (dalam Prayitno dan Safitri, 2017:2) “Database atau biasa disebut basis data merupakan kumpulan data yang saling berhubungan. Data tersebut biasanya terdapat dalam tabel-tabel yang saling berhubungan satu sama lain, dengan menggunakan field/kolom pada tiap tabel yang ada”.

Dapat disimpulkan bahwa Basis data adalah sistem komputerisasi yang berfungsi sebagai media penyimpanan data dan informasi saat dibutuhkan tanpa adanya pengulangan data yang tersimpan.

2.3.3 Pengertian PHP

Jannah dkk (2019:1), “Hypertext Preprocessor (PHP) merupakan bahasa pemrograman *script server-side* yang didesain untuk pengembangan web. PHP disebut bahasa pemrograman *server-side* karena diproses pada computer server.”

Menurut Sujatmiko (2012:213) “PHP (*Personal Home Page*) adalah bahasa pemrograman yang bisa kita gunakan untuk membuat aplikasi web”.

Dapat disimpulkan bahwa PHP adalah Bahasa pemrograman yang didesain untuk membuat ataupun mengembangkan sebuah website yang banyak fitur serta pemrograman yang lebih mudah.



Gambar 2.2 Logo PHP

(Sumber: <https://www.php.net/>)

2.3.4 Pengertian XAMPP

Menurut Kristanto(2018:29), “XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program”.



Sedangkan menurut Binarso dkk. (2017:76) “XAMPP adalah sebuah software web server Apache yang didalamnya sudah tersedia database server MySQL dan mendukung PHP Programming.”



Gambar 2.3 Logo XAMPP

(Sumber: <https://www.apachefriends.org/index.html>)

2.3.5 Pengertian MySQL

Menurut Jubilee (2018:2) “MySQL merupakan server yang melayani database. Untuk membuat dan mengolah database, kita dapat mempelajari pemrograman khusus yang disebut query (perintah) SQL.”

Sedangkan menurut Chandra dan Suroto (2018:19) “MySQL adalah sebuah software database yang merupakan tipe data relasional yang artinya MySQL menyimpan datanya dalam bentuk table-tabel yang saling berhubungan. MySQL adalah RDBMS (Relation Database Management System) yang cepat dan mudah digunakan, serta sudah banyak digunakan untuk berbagai kebutuhan.”

Kesimpulannya adalah MySQL adalah sistem manajemen database SQL yang bersifat open source dan paling populer saat ini. Sistem database MySQL mendukung beberapa fitur seperti multithreaded, multiuser dan SQL database Management System (DBMS).



Gambar 2.4 Logo MySQL

(Sumber: <https://www.mysql.com/>)