

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Membahas tinjauan pustaka dari sistem Presensi anggota atau karyawan lain dan membahas landasan teori yang berhubungan dengan sistem presensi anggota Polresta Surakarta.

Rancang bangun sistem absensi karyawan *online* berbasis web menggunakan *Framework* PHP Codeigniter & Mysql bertujuan untuk mengembangkan sistem informasi yang dapat digunakan sebagai sistem absensi agar karyawan disiplin dalam kerja dengan cepat dan akurat di di PT Strone Mitra Telekomunikasi. Membangun sistem absensi karyawan berbasis web dengan metode penelitian model *waterfall* (Satria, 2013).

Pembuatan sistem informasi absensi Karyawan bertujuan untuk menghasilkan sistem informasi absensi karyawan di PT. Harja Gunatama Lestari Bandung menggunakan *software Delphi 7*. Membangun sistem informasi absensi karyawan dengan menggunakan metode penelitian *waterfall* (Rinawati dan Candrawati, 2013).

Pembuatan Sistem Absensi Asisten Dosen bertujuan agar asisten dosen lebih disiplin dalam tugasnya. Membangun sistem informasi asisten dosen menggunakan *QR Code Scanner* berbasis android pada Program Studi Sistem Informasi Universitas Muria Kudus. Software yang digunakan Eclipse dan Dreamweaver CS6. Sistem absensi karyawan asisten dosen menggunakan *QR Code Scanner* (Nuddin dan Fithri, 2015).

Berdasarkan data dan kasus yang didapatkan, sehingga diperlukan untuk membuat sistem yang diharapkan dapat membantu Sie Propam Polresta Surakarta dalam merancang dan membuat Sistem Presensi Anggota Polresta Surakarta dengan menggunakan Bahasa Pemrograman PHP, MySQLi sebagai databasenya dan menggunakan jaringan wifi sebagai pembatas area presensi.

2.2 Kerangka Pemikiran

Kerangka pemikiran yang dijalankan dalam penelitian ini adalah sebagai berikut :

1) Latar belakang masalah

Tahapan paling awal, yakni menelusuri latar belakang kenapa masalah yang akan diangkat menjadi penting untuk dipilih.

2) Rumusan masalah

Penyimpulan latar belakang masalah menjadi suatu rumusan masalah yang akan diangkat untuk menjadi bahan penelitian.

3) Penguasaan dasar Bootstrap, CSS, PHP dan MySQLi

Tahap untuk mempelajari dasar-dasar Bootstrap, CSS, PHP dan MySQLi agar lebih menguasai program – program yang akan digunakan untuk membangun sistem.

4) Pengumpulan data tertulis dan tidak tertulis

Pengumpulan data dilakukan baik dengan tanya – jawab (*interview*), observasi, maupun studi literatur di perpustakaan.

5) Observasi aplikasi Bootstrap, CSS, PHP dan MySQLi

Merupakan tahap pengamatan sampel – sampel aplikasi yang telah ada, jurnal, buku, maupun karya ilmiah untuk kajian yang dapat dijadikan referensi untuk pembangunan sistem.

6) Analisis dan perancangan orientasi obyek

Implementasi Sistem Presensi Anggota Polresta Surakarta

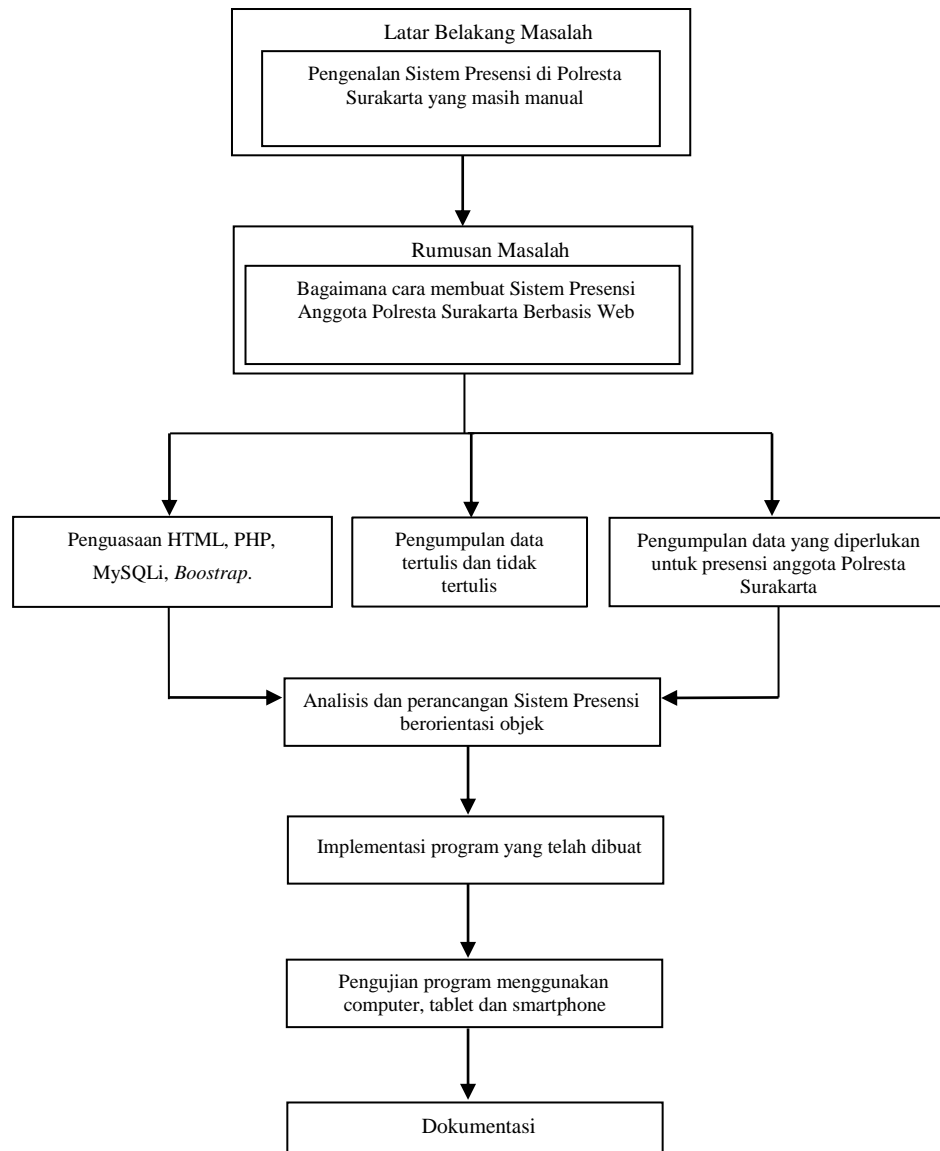
7) Pengujian Sistem

Pengujian sistem akan dilakukan pada beberapa komputer untuk mengetahui jika ada kesalahan dan kekurangan pada sistem.

8) Dokumentasi

Tahapan terakhir, yakni tahap pendokumentasian seluruh poses penyusunan tugas akhir ke dalam laporan.

Berdasarkan penjabaran dari kerangka pemikiran penelitian tentang sistem presensi Polresta Surakarta, maka dapat dibuatkan bagan alur seperti Gambar 2.1.



Gambar 2. 1 Kerangka Pemikiran

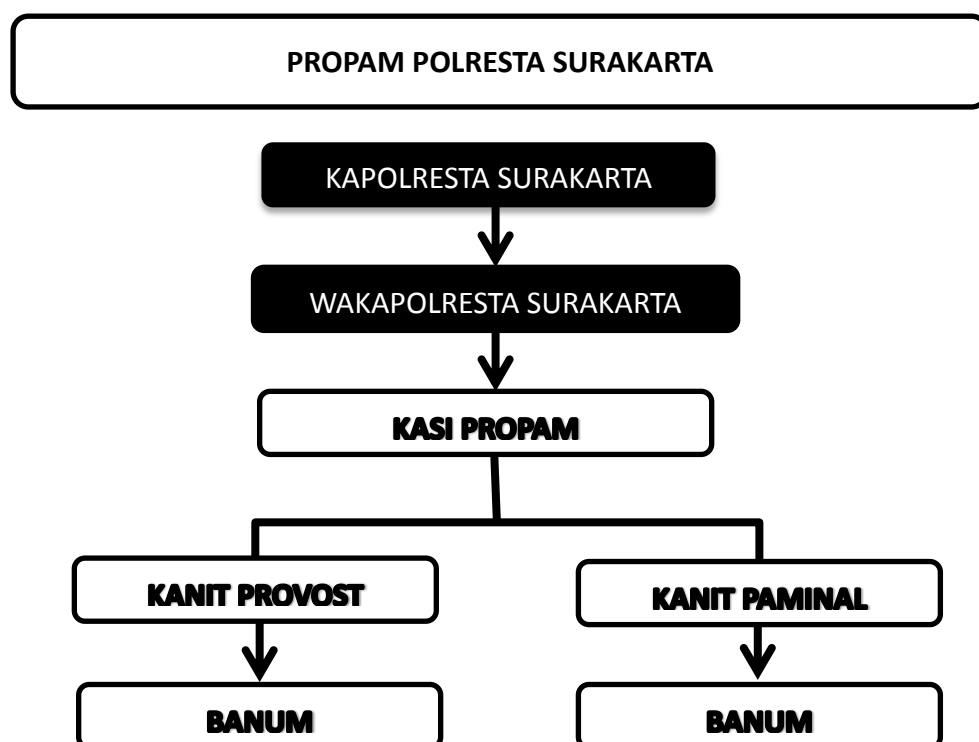
2.3 Landasan Teori

Beberapa istilah yang berhubungan dengan pembuatan membangun sistem presensi berbasis *web* anggota Polresta Surakarta.

2.3.1 Propam Polresta Surakarta

Seksi Profesi dan Pengamanan (Propam) berdasarkan Peraturan Kapolri nomor 23 tahun 2010 Pasal 28 yang selanjutnya disingkat Sipropam adalah unsur pengawas dan pembantu pimpinan di bidang provos dan pengamanan internal

pada tingkat Polres yang berada di bawah Kapolres dan bertugas melaksanakan pembinaan dan pemeliharaan disiplin, pengamanan internal, pelayanan pengaduan masyarakat yang diduga dilakukan oleh anggota Polri dan/atau PNS Polri, melaksanakan sidang disiplin dan/atau kode etik profesi Polri, serta rehabilitasi personel. Pada Pasal 29 menyebutkan Sipropam dipimpin oleh Kasipropam yang bertanggung jawab kepada Kapolres dan dalam pelaksanaan tugas sehari-hari di bawah kendali Wakapolres. Struktur organisasi Seksi Propam dapat dilihat pada Gambar 2.2



Gambar 2. 2 Struktur Seksi Spropam Polresta Surakarta

2.3.2 Sistem Presensi

Presensi adalah suatu kegiatan pencatatan terhadap setiap kehadiran karyawan dengan tujuan untuk mengetahui data yang berkaitan dengan kehadiran karyawan secara periodik baik harian maupun bulanan.

Sistem informasi kepegawaian secara umum digunakan sebagai sistem yang memungkinkan organisasi dalam membuat sebuah sistem terpusat yang memungkinkan pegawai dan manajer untuk mengakses informasi yang

berhubungan dengan pegawai. Ada beberapa manfaat yang dapat diperoleh melalui penerapan sistem informasi kepegawaian, yaitu dapat mengurangi biaya organisasi melalui otomatisasi kegiatan-kegiatan manual yang merupakan rutinitas pegawai, dapat menangani dan memfasilitasi komunikasi antar level di dalam organisasi, meningkatkan efisiensi, membantu mengelola beragam informasi berbeda mengenai pegawai di dalam organisasi, dan menyediakan sarana untuk membantu pengambilan keputusan yang berkaitan dengan manajemen sumber daya manusia. Sistem informasi kepegawaian mengubah data menjadi informasi yang diperlukan untuk operasional bisnis dan pengambilan keputusan (Wiblen, G dan Kristine, 2010).

2.3.3 Cuti dan Ijin

Cuti adalah keadaan untuk tidak masuk kerja yang diberikan dalam jangka waktu tertentu kepada Pegawai sesuai dengan ketentuan peraturan perundang-undangan.

Izin adalah keadaan untuk tidak masuk kerja yang diberikan kepada Pegawai untuk melaksanakan kepentingan pribadi atau dinas baik dilaksanakan di dalam negeri maupun di luar negeri dengan biaya sendiri atau sponsor.

Tujuan pengaturan Cuti dan Ijin sesuai Peraturan Kapolri nomor 11 tahun 2015 pasal 2, yaitu sebagai pedoman dalam pemberian Cuti dan Ijin kepada Pegawai dan terwujudnya tertib administrasi dan terpenuhinya hak Pegawai secara selektif untuk mendapatkan Cuti dan Ijin di lingkungan Polri.

Jenis cuti berdasarkan pasal 4 meliputi cuti tahunan, cuti sakit, cuti istimewa, cuti Ibadah keagamaan, cuti melahirkan, cuti di luar tanggungan Negara dan cuti karena alasan penting.

2.3.4 Pengenalan HTML5

Tahun 2011, HTML5 merupakan HTML versi terbaru mulai diperkenalkan dan berkembang terus menerus. *Browser* yang beredar saat ini juga masih beradaptasi dengan tag-tag HTML5 (Agung, 2016).

Berikut elemen yang dihapus dalam HTML5 :

- a. <acronym>
- b. <applet>
- c. <basefont>
- d. <big>
- e. <center>
- f. <dir>
- g.
- h. <frame>
- i. <frameset>
- j. <noframes>
- k. <strike>
- l. <tt>

2.3.5 *Bootstrap*

Bootstrap merupakan salah satu *framework* HTML, CSS dan JS yang cukup populer, serta banyak digunakan oleh para pengembang *web* saat ini. *Framework* ini banyak digunakan untuk membuat *website* bersifat *responsive*. Artinya bisa menyesuaikan tampilan *layout*-nya berdasarkan ukuran *viewport* dari *device* pengaksesnya, mulai dari *smartphone*, tablet, layar PC. Hal ini didasarkan pada fakta di lapangan yang menunjukkan bahwa perkembangan penggunaan *handheld* seperti *smartphone* dan tablet semakin meningkat. Dengan *website* responsif tersebut akan memudahkan pengguna melakukan *resizing*, *scrolling* dan *panning* dengan lebih optimal (Utomo, 2016).

2.3.6 Pengenalan PHP

PHP atau singkatan dari *Personal Home Page* merupakan bahasa skrip yang tertanam dalam HTML untuk dieksekusi bersifat server side". PHP termasuk dalam *open source product*, sehingga source code PHP dapat diubah dan didistribusikan secara bebas. Versi

terbaru PHP dapat diunduh secara gratis melalui situs resmi PHP : <http://www.php.net> (Arief, 2011).

PHP juga dapat berjalan pada berbagai *web* server seperti IIS (*Internet Information Server*), PWS (*Personal Web Server*), Apache, Xitami. PHP juga mampu berjalan di banyak sistem operasi yang beredar saat ini, diantaranya : Sistem Operasi Microsoft Windows (semua versi), Linux, Mac Os, Solaris. PHP dapat dibangun sebagai modul *web* server Apache dan sebagai binary yang dapat berjalan sebagai CGI (*Common Gateway Interface*). PHP dapat mengirim HTTP *header*, dapat mengatur *cookies*, mengatur *authentication* dan *redirect user*.

Salah satu keunggulan yang dimiliki PHP adalah kemampuannya untuk melakukan koneksi ke berbagai macam *software* sistem manajemen basis data atau *Database Management System* (DBMS), sehingga dapat menciptakan suatu halaman *web* dinamis. PHP mempunyai konektivitas yang baik dengan beberapa DBMS seperti *Oracle*, *Sybase*, *mSQL*, *MySQL*, *Microsoft SQL Server*, *Solid*, *PostgreSQL*, *Adabas*, *FilePro*, *Velocis*, *dBase*, *Unix dbm*, dan tidak terkecuali semua database ber-*interface* ODBC. Hampir seluruh aplikasi berbasis *web* dapat dibuat dengan PHP. Namun kekuatan utama adalah konektivitas basis data dengan *web*. Dengan kemampuan ini kita akan mempunyai suatu sistem basis data yang dapat diakses.

2.3.7 Pengertian Basis Data

Basis data adalah suatu kumpulan logikal data yang berhubungan dan dekripsi dari data tersebut yang di rancang untuk kebutuhan informasi suatu organisasi (Connolly dan Begg, 2010).

Basis data adalah kumpulan file yang saling terkait. Basis data tidak hanya merupakan kumpulan file. *Record* pada setiap file harus memperbolehkan hubungan-hubungan untuk menyimpan file lain. Keuntungan basis data yaitu :

- a. Kemampuannya untuk menggunakan data yang sama pada banyak aplikasi dan sistem.

- b. Penyimpanan data dalam format yang fleksibel. Hal ini didefinisikan secara terpisah dari sistem informasi dan program-program aplikasi yang akan menggunakan basis data.
- c. Teknologi basis data menyediakan *skalabilitas superior*, dalam arti basis data dan sistem yang menggunakannya dapat ditingkatkan atau dikembangkan untuk memenuhi kebutuhan-kebutuhan perubahan pada sebuah organisasi.
- d. Kemajuan independensi data yang sangat mengurangi redundansi data, telah meningkatkan fleksibilitas. Berdasarkan pengertian di atas, dapat disimpulkan basis data adalah sekumpulan data yang terintegrasi dan di rancang untuk memelihara informasi dan membuat informasi tersebut tersedia untuk memenuhi suatu kebutuhan organisasi.

2.3.8 MySQLi

Mysqli extension (*mysqli* merupakan singkatan dari *Mysql Improved*) pada dasarnya adalah perbaikan dari *mysql extension* dan dikembangkan untuk mendukung fitur-fitur terbaru untuk *Mysql* 4.1 ke atas. Akan tetapi, terhitung mulai dari *PHP* versi 5.5 *extension mysql* tidak lagi disarankan penggunaannya dan berstatus *deprecated*, yang berarti ada kemungkinan besar *mysql* akan dihapus pada versi *PHP* berikutnya.

Hampir semua fungsi yang ada pada *mysql extension* juga tersedia pada *mysqli*. *Syntax* (aturan penulisan) sangat mirip dengan *mysql extension* sehingga jika telah lama menggunakan *mysql extension* akan sangat mudah untuk beralih menggunakan *mysqli extension*.

Selain menambahkan huruf “i” di dalam nama fungsi, argumen-argumen yang dibutuhkan juga hampir mirip. Perbedaannya, jika dalam *extension mysql* pada umumnya meletakkan argumen *resource* di akhir fungsi, maka di dalam *mysqli*, argumen ini diletakkan di awal (Subagia, 2016).

2.3.9 UML

UML (*Unified Modeling Language*) adalah ‘bahasa’ pemodelan untuk sistem atau perangkat lunak yang berparadigma ‘berorientasi objek’. Pemodelan (modeling) sesungguhnya digunakan untuk penyederhanaan permasalahan-




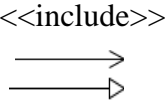
permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami.

Menurut Widodo (Widodo dan Herlawati, 2011), “Beberapa literatur menyebutkan bahwa UML menyediakan sembilan jenis diagram, yang lain menyebutkan delapan karena ada beberapa diagram yang digabung, misalnya diagram komunikasi, diagram urutan dan diagram pewaktuan digabung menjadi diagram interaksi”. Namun demikian model-model itu dapat dikelompokkan berdasarkan sifatnya yaitu statis atau dinamis.

2.3.7.1 Use Case Diagram

Use case diagram menampilkan sekumpulan *use case* dan aktor (pelaku) serta hubungan diantara *use case* dan aktor tersebut. *Use case diagram* digunakan untuk penggambaran *use case* static dari suatu sistem. *Use case* merupakan komponen wajib dalam program, dengan adanya *use case* pengguna dapat melihat gambaran dari kegunaan aplikasi tersebut. Adapun simbol dan fungsi simbol yang digunakan dalam *use case diagram* menurut (Sukamto dan Shalahuddin, Modul Pembelajaran Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek), 2011) dapat dilihat pada Tabel 2.1.

Tabel 2. 1 Simbol *Use Case Diagram*

GAMBAR	NAMA	KETERANGAN
	<i>Actor</i>	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri.
	<i>Use Case</i>	Fungsional yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.
	<i>Association</i>	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
	<i>Include</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.

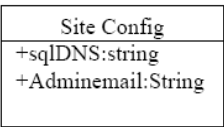

2.3.7.2 Class Diagram

Class diagram digunakan untuk menampilkan kelas-kelas dan paket-paket di dalam sistem. *Class diagram* memberikan gambaran untuk sistem secara statis dan relasi antar kelas. Umumnya, dibuat beberapa *class diagram* untuk sistem tunggal. Beberapa diagram akan menampilkan subset dari kelas-kelas dan relasinya. Dapat juga dibuat beberapa diagram sesuai dengan yang diinginkan untuk mendapatkan gambaran lengkap terhadap sistem yang dibangun.

Class diagram adalah alat perancangan terbaik untuk tim pengembang. Diagram tersebut membantu pengembang mendapatkan struktur sistem sebelum kode ditulis, dan membantu untuk memastikan bahwa sistem adalah desain terbaik.

Class diagram juga dapat diartikan sebagai kumpulan objek-objek dengan data yang mempunyai struktur umum, *behavior* umum, relasi umum, dan semantic atau kata yang umum. Sebuah *class* digambarkan seperti sebuah bujur sangkar dengan tiga bagian ruangan. *Class* sebaiknya diberi nama menggunakan kata benda sesuai dengan domain, bagian atau kelompoknya. Simbol class diagram dapat dilihat pada Tabel 2.2.

Tabel 2. 2 Simbol *Class Diagram*

GAMBAR	NAMA	KETERANGAN
	<i>Class</i>	<i>Class</i> adalah blok - blok pembangun pada pemrograman berorientasi obyek.
	<i>Association</i>	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .

2.3.7.3 Sequence Diagram


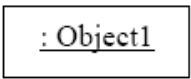


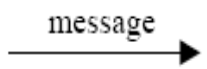
Sequence diagram merupakan salah satu jenis diagram pada UML yang menjelaskan interaksi objek yang berdasarkan urutan waktu. *Sequence diagram*

juga dapat menggambarkan urutan atau tahapan yang harus dilakukan untuk dapat menghasilkan sesuatu seperti pada *use case diagram*.

Sequence diagram digunakan untuk menggambarkan perilaku pada sebuah skenario. Diagram jenis ini memberikan kejelasan sejumlah objek dan pesan-pesan yang diletakkan diantaranya di dalam sebuah *use case*. Komponen utamanya adalah objek yang digambarkan dengan kota segi empat, *message* yang digambarkan dengan gari penuh, dan waktu yang ditunjukkan dengan *progress vertical*.

Secara mudahnya *sequence diagram* adalah gambaran tahap demi tahap yang seharusnya dilakukan untuk menghasilkan sesuatu sesuai dengan *use case diagram*. Simbol *Sequence Diagram* menurut (Sukamto dan Salahuddin, Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek, 2013) tersaji pada Tabel 2.3.




Tabel 2. 3 Simbol *Sequence Diagram*

GAMBAR	NAMA	KETERANGAN
	<i>Actor</i>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri.
	<i>Object</i>	Merupakan <i>instance</i> dari sebuah class dan dituliskan tersusun secara horizontal.
	<i>Lifeline</i>	Menyatakan kehidupan suatu objek.
	<i>Activation</i>	Menyatakan objek dalam keadaan aktif dan berinteraksi pesan. Semua yang berhubungan dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya.
	<i>Message</i>	Message mengindikasikan komunikasi antara object-object.

2.3.7.4 Activity Diagram

Activity diagram atau diagram aktivitas merupakan salah satu jenis diagram pada UML yang dapat memodelkan proses-proses apa saja yang terjadi pada sistem. *Activity diagram* juga dapat diartikan sebagai sebuah diagram yang menggambarkan alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi dan bagaimana akan berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. Simbol dan fungsi simbol yang digunakan dalam *activity diagram* menurut (Sukamto dan Salahuddin, Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek, 2013) dapat dilihat pada Tabel 2.4.

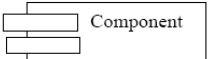

Tabel 2. 4 Simbol *Activity Diagram*

GAMBAR	NAMA	KETERANGAN
	Status Awal	Status awal aktivitas sistem
	Aktivitas	Aktivitas yang dilakukan sistem biasanya diawali dengan kata kerja.
	Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

2.3.7.5 Component Diagram

Component diagram dibuat untuk menunjukkan organisasi dan ketergantungan diantara kumpulan komponen dalam sebuah sistem. *Component diagram* fokus pada komponen sistem yang dibutuhkan dan ada di dalam sistem. Simbol Component Diagram dapat dilihat pada Tabel 2.5.

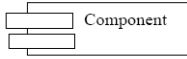
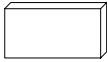

Tabel 2. 5 Simbol *Component Diagram*

GAMBAR	NAMA	KETERANGAN
	<i>Component</i>	Sebuah komponen melambangkan sebuah entitas software dalam sebuah sistem.
	<i>Dependency</i>	Sebuah Dependency digunakan untuk menotasikan relasi antara dua komponen.

2.3.7.6 Deployment Diagram

Deployment diagram merupakan salah satu diagram pada UML yang menunjukkan tata letak atau sistem secara fisik, dapat juga dikatakan untuk menampilkan bagian-bagian *software* yang terdapat *hardware* dan digunakan untuk menerapkan suatu sistem dan hubungan antara komponen *hardware*. *Deployment diagram* juga dapat diartikan sebagai salah satu jenis UML yang digunakan untuk memvisualisasikan, menspesifikan, dan mendokumentasikan proses yang terjadi pada suatu sistem perangkat lunak berbasis *object oriented* yang akan dibangun. Simbol Deployment Diagram dapat dilihat pada Tabel 2.6.

Tabel 2. 6 Simbol *Deployment Diagram*

GAMBAR	NAMA	KETERANGAN
	<i>Component</i>	komponen-komponen yang ada diletakkan di dalam node untuk memastikan keberadaan posisi mereka
	<i>Node</i>	Menggambarkan bagian-bagian <i>hardware</i> dalam sebuah sistem.
	<i>Association</i>	Menghubungkan dua <i>node</i> yang mengindikasikan jalur komunikasi antara elemen-emelen dengan <i>hardware</i>

2.3.10 Jaringan Komputer

Jaringan komputer adalah sistem yang terdiri atas beberapa unit komputer yang didesain sedemikian rupa sebagaimana tujuan utamanya yakni untuk dapat berbagi sumber daya (CPU, printer, scanner, plotter, harddisk, dan lain-lain), berkomunikasi (pesan instan, surel), dan dapat mengakses informasi (situs web) (Madcoms, 2015).

IP address dibagi ke dalam lima kelas, yaitu kelas A, kelas B, kelas C, kelas D dan kelas E. Perbedaan tiap kelas adalah pada ukuran dan jumlahnya. Penentuan kelas ini dilakukan dengan cara berikut :

- 1) Kelas A : digunakan untuk jaringan WAN, Ip address nya pada bagian pertama antara 0-127, dan yang merupakan Net ID nya yaitu 1 bagian yang pertama. Subnet mask nya 255.0.0.0 Contoh: 8.254.129.11
- 2) Kelas B : biasanya digunakan untuk jaringan MAN, Ip address nya pada bagian pertama antara 128-191, dan yang merupakan network ID nya yaitu 2 bagian pertama. Subnet masknya 255.255.0.0. Contoh: 128.255.129.7
- 3) Kelas C : biasanya digunakan untuk jaringan LAN, Ip address nya pada bagian pertama antara 192-223, dan yang merupakan network ID nya yaitu 3 bagian pertama. Subnet masknya 255.255.255.0. Contoh: 192.168.1.10
- 4) Kelas D : biasanya digunakan untuk keperluan multicasting. IP address nya pada bagian pertama antara 224-247. Dalam multicasting tidak dikenal network ID dan host ID.
- 5) Kelas E : biasanya digunakan untuk keperluan umum. IP address nya pada bagian pertama antara 248-255.

IP private adalah jenis IP yang saat akan menggunakannya tidak perlu diregistrasi sebab oleh router IP jenis ini tidak akan diteruskan. IP Private sesuai dengan kelasnya: yakni kelas A: 10.0.0.0-10.255.255.255, kelas B: 172.16.0.0-172.31.255.255 dan kelas C: 192.168.1.0-192.168.255.255

Sedangkan IP publik adalah jenis IP yang saat akan menggunakannya harus diregistrasi (ke badan penyalur IP address, dengan maksud supaya tidak bentrok) karena IP ini dapat melewati di lalu lintas jaringan internet melewati router-routernya.

2.3.11 Pengujian *Black Box*

Pengujian menggunakan sekumpulan aktifitas validasi, dengan pendekatan *black box testing*. Menurut (Sukamto dan Shalahuddin, Modul Pembelajaran Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek), 2011), *black box testing* adalah menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan. Pengujian kotak hitam dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan memakai perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan. Kasus uji yang dibuat untuk melakukan

pengujian *black box testing* harus dibuat dengan kasus benar dan kasus salah.

Kelebihan dari *black box testing* antara lain :

- a) Spesifikasi program dapat ditentukan di awal
- b) Dapat digunakan untuk menilai konsistensi program
- c) Testing dilakukan berdasarkan spesifikasi
- d) Tidak perlu melihat kode program secara detail.