

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Tinjauan Pustaka**

Menurut (Pangemanan, Anwariningsih, & Susilo, 2013) Peran informasi semakin dibutuhkan untuk mendukung suatu kegiatan. Sejalan dengan perkembangan teknologi informasi baik hardware maupun software yang meningkat pesat, maka seharusnya sebuah institusi dapat memanfaatkan kemajuan teknologi informasi. Salah satu bentuk pemanfaatannya adalah untuk pengembangan sistem informasi yang dapat menunjang aktivitas karyawan dalam mengambil keputusan. Dengan sistem ini, aktivitas karyawan di suatu institusi dapat optimal dan efisien sehingga dapat meningkatkan produktivitas. Kegiatan Terpadu Sistem Informasi Pegawai Universitas Sahid Surakarta merupakan perangkat lunak kegiatan pengolahan data karyawan dan dosen Universitas Sahid Surakarta. Fungsi yang setiap kegiatan di Universitas Sahid Surakarta baik di luar maupun di dalam kampus dapat dikendalikan sehingga dapat meminimalisir

Menurut (Noviyanti, Retnoningsih, & Saputro Al Haris, 2018) Proses seleksi yang efektif harus dilakukan dengan mempertimbangkan informasi tentang analisis pekerjaan, karena dalam analisis pekerjaan itu tergambar uraian pekerjaan yang akan dilakukan. Berbagai persyaratan yang harus dipenuhi oleh para karyawan yang melakukan pekerjaan tersebut. Akan tetapi dalam proses tersebut terkadang tidak berjalan dengan semestinya dan hasil yang dicapai kemungkinan tidak sesuai dengan target awal PT. Datacom Solusindo mempunyai supplier yang selalu menawarkan berbagai produk untuk perusahaan.

Pelaksanaannya perusahaan membutuhkan karyawan yang kompeten untuk menjalankan segala aktifitas perusahaan. Sehingga perusahaan melakukan perekrutan dan seleksi karyawan sesuai dengan bidang yang dibutuhkan. Perusahaan sudah menetapkan standart bagi calon para karyawan yang ingin masuk di perusahaan. Dengan tujuan penelitian Membangun suatu aplikasi

Sistem Seleksi Calon Karyawan yang diharapkan dapat membantu PT. Datacom Solusindo dalam penyeleksian calon karyawan baru.

Menurut (Anjarwani & Khusnuliawati, 2018) Rumah Sakit Umum Daerah (RSUD) Kota Surakarta merupakan sebuah instansi pemerintahan yang bergerak di bidang kesehatan. RSUD Kota Surakarta sebelumnya sudah memiliki sistem aplikasi seperti Sistem Rawat Inap dan Sistem Rawat Jalan, RSUD Kota Surakarta juga memiliki sistem yang terpusat dari Pemerintah Kota Surakarta seperti sistem absensi, namun karena RSUD Kota Surakarta ini merupakan instansi yang baru berkembang sehingga masih banyak sistem aplikasi yang masih dijalankan secara manual salah satunya pada Sistem Informasi Pegawai Negeri Sipil.

Berdasarkan permasalahan tersebut maka penelitian ini membuat Sistem Informasi Pegawai Negeri Sipil berbasis *website* yang berisi tentang data-data Pegawai Negeri Sipil yang ada di RSUD Kota Surakarta. Sistem informasi pegawai negeri sipil di RSUD Kota Surakarta berbasis *website* bertujuan untuk memudahkan admin/pegawai tata usaha untuk mendata karyawan di RSUD Kota Surakarta, selain itu *website* ini berfungsi sebagai media informasi kepada pasien, pengunjung maupun masyarakat umum.

## **2.2 Kerangka Pemikiran**

Berikut ini adalah tahapan kerangka pemikiran yang akan dijalankan oleh penulis dalam membangun Sistem Informasi Karyawan di CV. Jaya Mandiri pada Gambar 2.1.

Uraian dari kerangka berfikir sebagai berikut :

### **1. Latar Belakang Masalah**

Latar belakang masalah pada tugas akhir ini adalah belum adanya sistem informasi karyawan, yang masih menggunakan pembukun untuk menyimpan identitas karyawan, sehingga untuk mencari data karyawan yang sudah lama bekerja atau yang sudah keluar akan memerlukan waktu yang cukup lama.

2. Rumusan Masalah

Rumusan masalah pada tugas akhir ini adalah Bagaimana cara membuat Sistem Informasi Karyawan di CV. Jaya Mandiri yang dapat memberikan informasi.

3. Pengumpulan Data

Pengumpulan data dalam penelitian ini ada dua yaitu pengumpulan data tertulis dan pengumpulan data tidak tertulis. Penelitian ini mengumpulkan semua data yang dibutuhkan dengan melakukan *interview* dengan bagian umum di Universitas Sahid Surakarta.

4. Penguasaan Dasar

Penguasaan dasar dalam penelitian ini adalah penguasaan yang telah diketahui atau dikuasai penulis mengenai *PHP* dan *MySQL*.

5. Observasi Aplikasi

Observasi alat dalam penelitian ini adalah penguasaan terhadap *software-software* yang akan digunakan untuk membuat sistem informasi data karyawan di CV. Jaya Mandiri.

6. Analisis dan Perancangan Sistem

Analisis dan perancangan sistem dalam penelitian ini adalah memahami dengan detail, apa yang harus dilakukan untuk sistem yang akan dibuat serta merancang sistem yang baru.

7. Implementasi Sistem

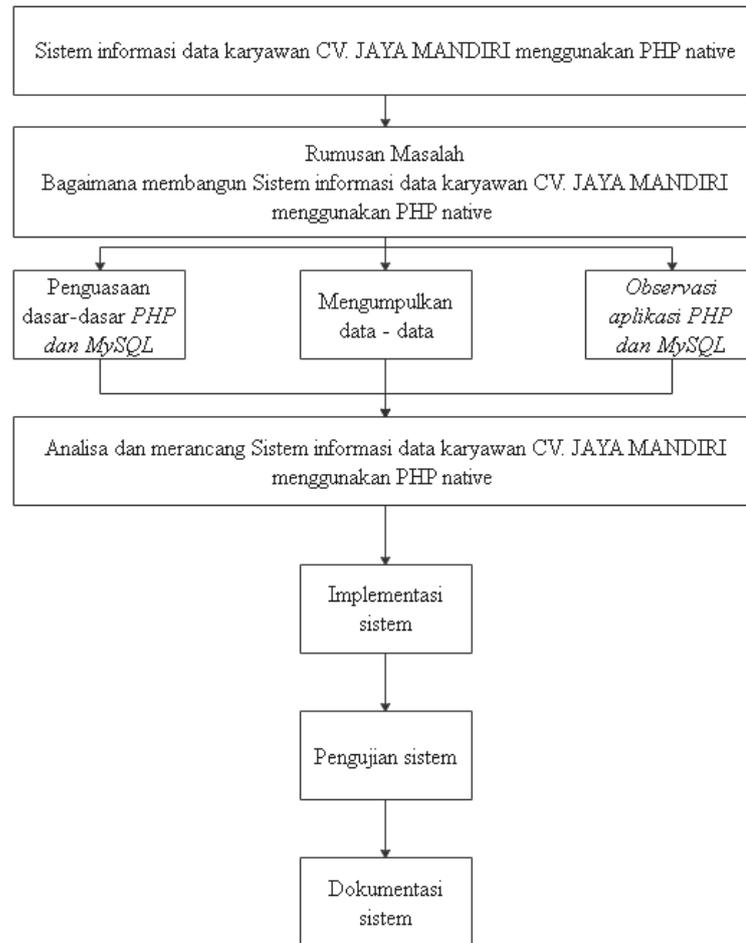
Implementasi sistem dalam penelitian ini adalah proses pelaksanaan penelitian, setelah dilakukan analisis dan perancangan kemudian dilakukan pembuatan sistem.

8. Pengujian Sistem

Pengujian sistem dalam penelitian ini adalah dilakukan untuk mengetahui jika ternyata masih ada kesalahan atau kekurangan pada sistem yang dibuat.

9. Penerapan Sistem dan Dokumentasi

Penerapan sistem dan dokumentasi dalam penelitian ini adalah proses menerapkan aplikasi di CV. Jaya Mandiri dan proses pengambilan dokumentasi setelah sistem selesai dibuat dari keseluruhan kegiatan.



Gambar 2.1 Kerangka Pemikiran

## 2.3 Dasar Teori

### 2.3.1 Website

*Website* merupakan kumpulan halaman-halaman yang berisi informasi yang disimpan di internet yang bisa diakses atau dilihat melalui jaringan internet pada perangkat-perangkat yang bisa mengakses internet itu sendiri seperti komputer. Definisi kata *web* adalah *web* sebenarnya penyederhanaan dari sebuah istilah dalam dunia komputer yaitu *World Wide Web* yang merupakan bagian dari teknologi internet (Hastanti, Purnama, & Wardati, 2015).

*World Wide Web* atau disingkat dengan nama *www*, adalah sebuah bagian dari internet yang sangat dikenal dalam dunia internet, dengan adanya *www* seorang pengguna dapat menampilkan sebuah halaman *virtual* yang disebut dengan *website* (Nugroho, 2014).

Internet dapat diartikan sebagai jaringan komputer yang luas dan besar yang mendunia, yaitu menghubungkan pemakai komputer dari negara ke negara di seluruh dunia. Pada awalnya internet atau *web* hanya dipergunakan untuk kepentingan militer yaitu suatu teknologi yang dipergunakan untuk mengirimkan pesan melalui satelit. Akan tetapi lama kelamaan teknologi tersebut akhirnya meluas, dan bahkan internet pada saat ini sudah sama populernya dengan *telephone*. Informasi yang dikirimkan lewat internet dapat diakses ke seluruh dunia hanya dalam hitungan menit bahkan detik (Hastanti, Purnama, & Wardati, 2015).

Teknologi yang digunakan menjadi sangat populer dan cepat sekali perkembangannya. Saat ini internet sudah tidak menjadi istilah yang asing lagi. Suatu informasi yang dikirimkan lewat internet dapat berupa teks, gambar maupun multimedia sehingga internet juga dimanfaatkan oleh perusahaan-perusahaan untuk mempromosikan produk-produknya dengan cepat dan mudah.

### **2.3.2 Pemrograman**

*PHP* merupakan bahasa pemrograman pelengkap *HTML (Hypertext Markup Language)* yang memungkinkan aplikasi *web* dinamis untuk pengolahan data, pemrosesan data dari *user* via *form*, membuat buku tamu, toko *online*, dan lain sebagainya, dengan mudah *PHP* dapat melakukan koneksi ke *database* karena *PHP* memang dilengkapi fitur yang memungkinkan koneksi ke *PHP* dilakukan dengan mudah, tanpa harus melakukan pemrograman yang memusingkan. *PHP* juga merupakan bahasa pemrograman berbasis *server side* yang dapat melakukan parsing *script PHP* menjadi *script web* sehingga dari sisi *client* menghasilkan suatu tampilan yang menarik (Ardhana, 2014).

Jadi dapat disimpulkan bahwa pengertian *Personal Home Page (PHP)* adalah bahasa pemrograman pelengkap *HTML* berbasis *server side* yang memungkinkan aplikasi *web* dinamis, dapat melakukan koneksi ke *database* dan menghasilkan suatu tampilan yang menarik (Rahmawati, 2017).

*HTML* adalah sebuah bahasa markah yang digunakan untuk membuat sebuah halaman *web*, menampilkan berbagai informasi di dalam sebuah penjelajah *web* internet dan pemformatan *hypertext* sederhana yang ditulis dalam berkas

format *ASCII* agar dapat menghasilkan tampilan wujud yang terintegrasi. *HTML* adalah bahasa *markup* untuk menstrukturkan dan menampilkan isi dari *World Wide Web*, sebuah teknologi inti dari internet.

Jadi dapat disimpulkan bahwa pengertian *Hypertext Markup Language* (*HTML*) adalah sebuah bahasa *markup* untuk membuat sebuah halaman *web* yang menampilkan berbagai informasi untuk menstrukturkan dan menampilkan isi dari *World Wide Web* pada *browser* yang ditulis dalam berkas format *ASCII* (Rahmawati, 2017).

### 2.3.3 Basis Data

Berbagai definisi tentang *MySQL* yang dikutip dari beberapa sumber yang dapat dilihat pada Tabel 2.1 untuk memberikan penjelasan lebih lanjut tentang *MySQL* (Rahmawati, 2017).

Tabel 2.1 Definisi *MySQL*

Sumber	Definisi
Faizal & Irnawati (2015:4)	<i>MySQL</i> adalah perangkat lunak sistem manajemen basis data <i>SQL</i> (bahasa Inggris : <i>database management system</i> ) atau <i>DBMS</i> yang <i>multithread</i> , <i>multi-user</i> . <i>MySQL</i> menggunakan <i>SQL</i> ( <i>Structure Query Language</i> ) sebagai bahasa dasar untuk mengakses <i>database</i> .
Raharjo (2015:16)	<i>MySQL</i> merupakan <i>software RDBMS</i> (atau <i>server database</i> ) yang dapat mengelola <i>database</i> dengan sangat cepat, dapat menampung data dalam jumlah sangat besar, dapat diakses oleh banyak <i>user</i> ( <i>multi-user</i> ) dan dapat melakukan suatu proses secara sinkron atau berbarengan ( <i>multi-threaded</i> ).
Nugroho (2014: 31)	<i>MySQL</i> adalah <i>software</i> atau program aplikasi <i>database</i> , yaitu <i>software</i> yang dapat dipakai untuk menyimpan data berupa informasi, teks dan juga angka.
Ardhana (2014:46)	<i>MySQL</i> adalah sebuah perangkat lunak sistem manajemen basis data <i>SQL</i> ( <i>database management system</i> ) atau <i>DBMS</i> yang <i>multithread</i> , dan <i>multi-user</i> .

Dari berbagai uraian pada Tabel 2.1 dapat disimpulkan bahwa pengertian *MySQL* adalah sebuah perangkat lunak sistem manajemen basis data *SQL* atau *DBMS software* yang dapat dipakai untuk menyimpan data berupa informasi, teks dan juga angka.

*Xampp* merupakan paket PHP yang berbasis *Open Source* yang dikembangkan oleh sebuah komunitas *Open Source*. Penggunaan perangkat lunak *XAMPP* diawali dengan *install* paket *XAMPP* pada halaman resmi. Tersedia beberapa *update* yang dapat di-*download* sesuai dengan *platform* komputer pengguna. Setelah penginstalan selesai maka pengguna dapat memulai pemrograman dengan membuka *XAMPP Control Panel* terlebih dahulu untuk mengaktifkan *service* yang disediakan seperti : *Apache, MySQL, FileZilla, Mercury dan Tomcat* dengan mengklik *Action : Star* (Rahmawati, 2017).

#### **2.3.4 Bootstrap**

*Bootstrap* sebuah alat bantu untuk membuat sebuah tampilan halaman *website* yang dapat mempercepat pekerjaan seorang pengembang *website* ataupun pendesain halaman *website*. Sesuai namanya, *website* yang dibuat oleh alat bantu ini memiliki tampilan halaman yang sama atau mirip dengan tampilan halaman *twitter* atau desainer juga dapat mengubah tampilan halaman *website* sesuai dengan kebutuhan. Tampilan *website* yang dibuat *bootstrap* akan menyesuaikan ukuran layar dari *browser* yang kita gunakan baik *dekstop, tablet* ataupun *mobile device*. Fitur ini bisa diaktifkan ataupun di-*non*-aktifkan sesuai keinginan. Sehingga, membuat *web* untuk tampilan *dekstop* saja dan apabila di-*reader* oleh *mobile browser* maka tampilan dari *web* yang dibuat sesuai layar. Dengan *bootstrap* kita juga bisa membangun *web* dinamis ataupun statis (Widyantoro, 2014).

#### **2.3.5 UML**

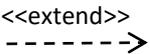
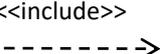
Pada perkembangan teknologi perangkat lunak diperlukan adanya bahasa yang digunakan untuk memodelkan perangkat lunak yang akan dibuat dan perlu adanya standarisasi agar orang di berbagai negara dapat mengerti pemodelan perangkat lunak. Pada perkembangan teknik pemrograman berorientasi objek, muncullah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat

lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu *Unified Modelling Language* (Salahudin dan Sukamto, 2016).

### 2.3.8.1 Use Case Diagram

*Use case* atau *diagram use case* merupakan pemodelan untuk melakukan sistem informasi yang akan dibuat. *Diagram use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. (Salahudin dan Sukamto, 2016). Simbol-simbol yang ada pada diagram *use case* dapat dilihat pada Tabel 2.2.

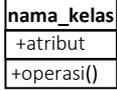
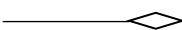
Tabel 2.2 Simbol-simbol *Use Case Diagram*

NO.	SIMBOL	NAMA	KETERANGAN
1.		<i>Use case</i>	Fungsional yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan kata kerja di awal di awal frase nama <i>use case</i> .
2.		<i>Actor</i>	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.
3.		<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.
4.		<i>Extend</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu.
5.		<i>Generalizati on</i>	Hubungan generalisasi dan spesialisasi (umum – khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
6.		<i>Include</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.

### 2.3.8.2 Class Diagram

Diagram kelas atau *class* diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas diagram memiliki atribut dan metode operasi. (Salahudin dan Sukanto, 2016). Simbol-simbol yang ada pada *class diagram* dapat dilihat pada Tabel 2.3.

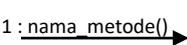
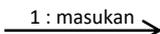
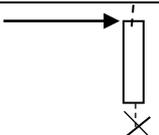
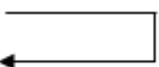
Tabel 2.3 Simbol-simbol *Class Diagram*.

NO.	SIMBOL	NAMA	KETERANGAN
1.		<i>Class</i>	Kelas pada struktur sistem
2.		<i>Interface</i>	Sama dengan konsep interface dalam pemrograman berorientasi objek.
3.		<i>Association</i>	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
4.		<i>Directed association</i>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain biasanya juga disertai dengan <i>multiplicity</i> .
5.		<i>Generalization</i>	Relasi antar kelas dengan makna generalisasi - spesialisasi ( umum – khusus).
6.		<i>Dependency</i>	Relasi antar kelas dengan makna kebergantungan antar kelas.
7.		<i>aggregation</i>	Relasi antar kelas dengan makna semua bagian ( <i>whole-part</i> ).

### 2.3.8.3 Sequence Diagram

Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek (Salahudin dan Sukanto, 2016). Simbol-simbol yang ada pada *sequence* diagram dapat dilihat pada Tabel 2.4.

Tabel 2.4 Simbol-simbol *Sequence Diagram*.

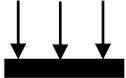
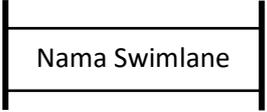
NO.	SIMBOL	NAMA	KETERANGAN
1.		Aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
2.		<i>Lifeline</i>	Menyatakan kehidupan suatu objek.
3.		Objek	Menyatakan objek yang berinteraksi pesan.
4.		Waktu aktif	Menyatakan objek dalam keadaan aktif dan berinteraksi.
5.		Pesan tipe create	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat
6.		Pesan tipe call	Menyatakan suatu objek memanggil operasi / metode yang ada pada objek lain atau dirinya sendiri.
7.		Pesan tipe send	Menyatakan bahwa suatu objek mengirimkan data / masukan / informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
8.		Pesan tipe <i>destroy</i>	Menyatakan suatu objek mengakhiri hidup objek lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i> .
9.		Pesan tipe <i>return</i>	Menyatakan bahwa suatu objek telah menjalankan suatu operasi atau metode menghasilkan suatu pengembalian ke objek tertentu, arah panah ke objek kembalian

#### 2.3.8.4 Activity Diagram

Diagram aktivitas atau *Activity diagram* menggambarkan *workflow* atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini diagram aktivitas menggambarkan aktivitas

sistem bukan apa yang dilakukan aktor (Salahudin dan Sukamto, 2016). Simbol-simbol yang ada pada *activity* diagram dapat dilihat pada Tabel 2.5.

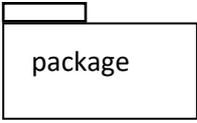
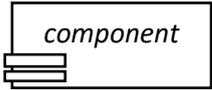
Tabel 2.5 Simbol-simbol *Activity Diagram*.

NO	GAMBAR	NAMA	KETERANGAN
1		Status awal	Status awal aktivitas sistem.
2		Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3		<i>Decision</i>	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4		<i>Join</i>	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5.		Status akhir	Status akhir yang dilakukan sistem.
6.		<i>Swimlane</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.
7.		<i>Fork</i>	Asosiasi percabangan dimana lebih dari satu aktivitas cabangkan menjadi satu.
8.		<i>Note</i>	Note adalah diagram diagram yang tidak memiliki pengaruh semantik pada elemen model.

### 2.3.8.5 Component Diagram

Diagram komponen *atau component* diagram dibuat untuk menunjukkan organisasi dan ketergantungan diantara kumpulan komponen dalam sebuah sistem (Salahudin dan Sukamto, 2016). Simbol-simbol yang ada pada *component* diagram dapat dilihat pada Tabel 2.6.

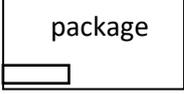
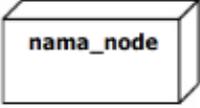
Tabel 2.6 Simbol – simbol *Component Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1.		<i>package</i>	<i>Package</i> merupakan sebuah bungkusan dari satu atau lebih komponen
2.		<i>Component</i>	Komponen sistem
3.		<i>Dependency</i>	Kebergantungan antar komponen, arah panah mengarah pada komponen yang dipakai
4.		<i>interface</i>	Sama dengan konsep <i>interface</i> pada pemrograman berorientasi objek, yaitu sebagai antarmuka komponen agar tidak mengakses langsung komponen
5.		<i>Link</i>	Relasi antar komponen

### 2.3.8.6 Deployment Diagram

*Deployment* diagram menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi. Diagram *deployment* juga dapat digunakan untuk memodelkan sistem tambahan, sistem *client*, sistem terdistribusi murni, rekayasa ulang aplikasi (Salahudin dan Sukamto, 2016). Simbol-simbol yang ada pada *deployment* diagram dapat dilihat pada Tabel 2.7.

Tabel 2.7 Simbol-Simbol *Deployment Diagram*

NO	SIMBOL	NAMA	KETERANGAN
1.		<i>Package</i>	<i>Package</i> merupakan sebuah bungkusan dari satu atau lebih <i>node</i> .
2.		<i>Node</i>	Biasanya mengacu pada perangkat keras ( <i>hardware</i> ), perangkat lunak yang tidak dibuat sendiri ( <i>software</i> ), jika didalam <i>node</i> disertakan komponen untuk mengkonsistenkan rancangan maka komponen yang diikutsertakan harus sesuai dengan komponen yang telah didefinisikan sebelum pada diagram komponen.
3.		<i>Dependency</i>	Kebergantungan antar <i>node</i> , arah panah mengarah pada <i>node</i> yang dipakai.
4.		<i>Link</i>	Relasi antar <i>node</i> .

### 2.3.6 Sistem Informasi Manajemen Sumber Daya Manusia

Sistem Informasi Manajemen pada dasarnya berhubungan dengan pemrosesan data menjadi suatu informasi. Koleksi data ini melibatkan penggunaan Teknologi Informasi, yaitu : komputer dan jaringan telekomunikasi (email, internet, telepon, dan teknologi informasi lainnya). Komputer sangat penting digunakan untuk pengumpulan data kuantitatif dan kualitatif. Kemampuan komputer dalam menyimpan data juga sangat membantu dalam pembuatan data menjadi suatu informasi. Dengan adanya Sistem Informasi Manajemen akan memampukan bisnis untuk menyediakan jawaban kepada manajer dalam menyediakan informasi dan pengetahuan yang terkandung di dalamnya (Satyanarayana dkk, 2009).

Sistem Informasi Manajemen Sumber Daya Manusia mengacu pada system dan proses yang ditempatkan pada pertemuan antara Sistem Informasi dengan

menggunakan Teknologi Informasi dan Komunikasi dan manajemen Sumber Daya Manusia (Edelhauser, 2012).

Suatu departemen sumber daya manusia memiliki begitu banyak data yang mencakup ketenagakerjaan yang ada dalam lingkungan perusahaan. Oleh karena itu, hal ini dapat menjadi suatu masukan terhadap Sistem Informasi Sumber Daya Manusia. Sistem Informasi Sumber Daya Manusia diharapkan dapat melakukan pengolahan terhadap data yang ada menjadi suatu informasi yang dapat di analisa dan mendukung pengambilan suatu keputusan.

### **2.3.7 Black Box Testing**

*Black-Box Testing* (*pengujian kotak hitam*) yaitu 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 kotak hitam harus dibuat dengan benar dan kasus salah, misalkan untuk kasus proses login maka kasus uji yang dibuat adalah :

1. Jika *user* memasukkan nama pemakai (*username*) dan kata sandi (*password*) yang benar.
2. Jika *user* memasukkan nama pemakai (*username*) dan kata sandi (*password*) yang salah, misalnya nama pemakai benar tapi kata sandi salah, atau sebaliknya, atau keduanya salah.

Pengujian *black-box* berfokus pada persyaratan fungsional perangkat lunak. Pengujian *black-box* memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi *input* yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. Pengujian *black-box* bukan merupakan alternatif dari teknik *white-box*, tetapi merupakan pendekatan komplementer yang kemungkinan besar mampu mengungkap kelas kesalahan daripada metode *white-box*. Pengujian *black-box* berusaha menemukan kesalahan dalam kategori sebagai berikut: (1) fungsi-fungsi yang tidak benar atau hilang, (2) kesalahan *interface*, (3)

kesalahan dalam struktur data atau akses *database external*, (4) kesalahan kinerja, (5) inisialisasi dan kesalahan terminasi

Tidak seperti pengujian *white-box* yang dilakukan pada saat awal proses pengujian, pengujian *black-box* cenderung diaplikasikan selama tahap akhir pengujian. Karena pengujian *black-box* memperhatikan struktur kontrol, maka perhatian berfokus pada *domain* informasi. Pengujian di desain untuk menjawab pertanyaan-pertanyaan berikut :

1. Bagaimana validitas fungsional diuji ?
2. Kelas *input* apa yang akan membuat *test case* menjadi baik ?
3. Apakah sistem sangat sensitif terhadap harga *input* tertentu ?
4. Bagaimana batasan dari suatu data diisolasi ?
5. Kecepatan data apa dan volume data apa yang dapat ditolelir oleh sistem ?
6. Apa pengaruh kombinasi tertentu dari data terhadap operasi sistem ?

Mengaplikasikan teknik *black-box*, maka serangkaian *test case* akan memenuhi kriteria berikut ini : (1) *test case* yang mengurangi, dengan harga lebih dari satu, jumlah *test case* tambahan yang harus didesain untuk mencapai pengujian yang dapat dipertanggungjawabkan, dan (2) *test case* yang memberi tahu mengenai kehadiran atau ketidakhadiran kelas kesalahan, daripada memberitahu kesalahan yang berhubungan hanya dengan pengujian spesifik yang ada (Pressman, 2002).