

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Penelitian ini dilakukan tidak terlepas dari hasil penelitian terdahulu yang pernah dilakukan. Penelitian terdahulu digunakan sebagai bahan kajian dan perbandingan. Penelitian – penelitian yang digunakan masih ada kaitannya dengan pembuatan rancang bangun aplikasi kasir toko aksesoris Samsung Cabang Surakarta.

Penelitian dari Pribadi (2015) Sistem Informasi Penjualan Pakaian di Distro Planet Clothing Berbasis *Web*. Masalah yang terjadi antara lain jarang ada pelanggan, dan laporan tidak lengkap. Tujuan dari penelitian adalah membangun sebuah Sistem Informasi Penjualan Pakaian di Distro Planet Clothing Berbasis *Web* yang akan digunakan Distro Planet Clothing. Sistem ini diharapkan dapat memperbanyak pelanggan pakaian pada Distro Planet Clothing, mempermudah dalam pengolahan data serta mempercepat transaksi. Sistem ini dibuat dengan PHP dan MySQL sebagai basis datanya.

Penelitian dari Cardoso (2013) menulis tentang Sistem Penjualan Helm Berbasis Web di Toko Elamor. Toko Elamor memiliki banyak konsumen sehingga sering kesulitan apabila ada pesanan dalam jumlah besar, selain itu terdapat kesulitan pengecekan stok barang. Dalam hal keuangan, sering terjadi kesalahan dalam penghitungan keuangan karena sistem masih manual. Tujuan sistem yang dibuat akan menyajikan informasi yang lebih cepat dan akurat untuk mendukung berbagai pihak dalam pengambilan keputusan. Kelemahan pada sistem ini adalah tidak menerima pengembalian barang. Sistem ini dibuat dengan PHP dan MySQL sebagai basis datanya.

Penelitian dari Laksono (2015) Sistem Penjualan Sepeda Secara *Online* pada Toko Berkat Bike. Toko Berkat Bike masih melakukan transaksi penjualan secara manual sehingga sering terjadi masalah pada saat pembuatan laporan. Masalah yang terjadi antara lain data yang tidak akurat, jarang ada pelanggan, data laporan tidak

lengkap. Tujuan dari penelitian adalah membangun sebuah sistem informasi penjualan sepeda secara *online* yang akan digunakan toko Berkat Bike. Sistem informasi ini dibuat dengan PHP dan MySQL sebagai basis datanya. Berdasarkan tinjauan pustaka diatas yang di harapkan memberikan refrensi dalam penlitian rancang bangun aplikasi kasir toko aksesoris samsung Cabang Surakarta yang di harapkan dapat meringan kan petugas kasir untuk proses transaksi dan pembuatan laporan. Pada pembuatan aplikasi kasir ini dibuat dengan PHP dan MySQL sebagai basis datanya.

2.2 Landasan Teori

2.2.1 Sistem

Perangkat lunak aplikasi adalah suatu subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna. Biasanya dibandingkan dengan perangkat lunak sistem yang mengintegrasikan berbagai kemampuan komputer, tapi tidak secara langsung menerapkan kemampuan tersebut untuk mengerjakan suatu tugas yang menguntungkan pengguna. (Hastanti, R. P. 2015)

2.2.2 Sistem Informasi

Penjualan merupakan tujuan dari pemasaran artinya perusahaan melalui departemen / bagian pemasaran termasuk tenaga penjualan (*sales force*) nya akan berupaya melakukan kegiatan penjualan untuk menghabiskan produk yang dihasilkan. (Hastanti, R. P. 2015)

2.2.3 Toko Aksesoris Samsung

Berawal dari toko yang hanya menjual produk *handphone* Samsung, toko yang terletak di pusat kota surakarta yang bertempat di Solo Grand Mall Lt.3 Blok M3 ini sekarang juga menyediakan berbagai macam aksesoris original mulai dari aksesoris luar maupun dalam *handphone* (*sparepart*).

Toko aksesoris ini berdiri di Kota Surakarta tergolong sangat baru yaitu pada tahun 2015, yang sebelumnya di Kota Surakarta ini hanya menjual *handphone* Samsung tetapi mengingat penjualan dan pengguna yang banyak oleh karena itu di PT

Samsung *Electronic* Indonesia menyediakan tempat yang menyediakan berbagai kebutuhan aksesoris dan juga layanan *service center* yang dianggap akan mencukupi kebutuhan dari pengguna dan pelanggan.

2.2.4 PHP

PHP kepanjangan dari *Hypertext Preprocessor*. PHP merupakan suatu bahasa skrip yang dapat ditanamkan atau disisipkan ke dalam HTML. PHP banyak dipakai untuk memprogram situs *website* dinamis. PHP dapat digunakan untuk membangun sebuah CMS.

PHP merupakan secara umum dikenal dengan sebagai bahasapemrograman script – script yang membuat dokumen HTML secara *on thefly* yang dieksekusi di *server* web, dokumen HTML yang dihasilkan dari suatu aplikasi bukan dokumen HTML yang dibuat dengan menggunakan editor teks atau editor HTML, dikenal juga sebagai bahasa pemrograman *server side* (Buana, 2014).

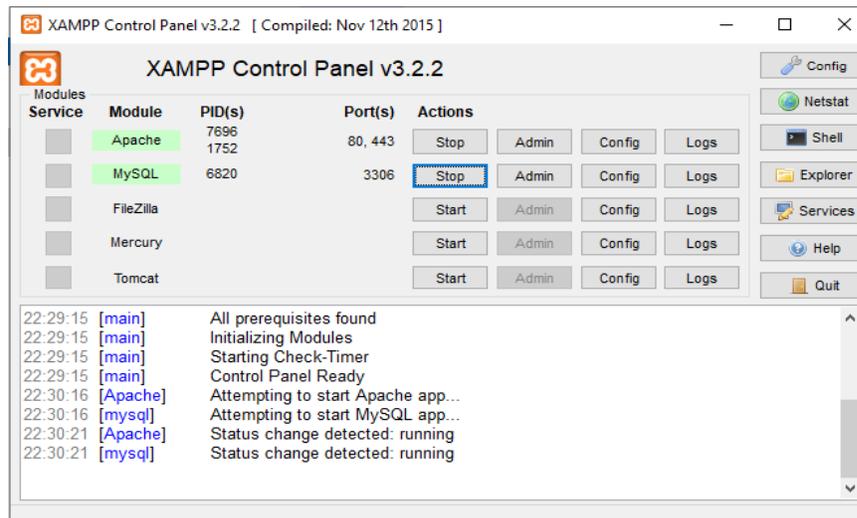
2.2.5 MySQL

MySQL adalah sebuah *server database open source* yang terkenal yang digunakan berbagai aplikasi terutama untuk server atau membuat *website*. MySQL berfungsi sebagai SQL (*Structured Query Language*) yang dimiliki sendiri dan sudah diperluas oleh MySQL umumnya digunakan bersamaan dengan PHP untuk membuat aplikasi *server* yang dinamis.

MySQL merupakan *software database* yang termasuk paling populer di lingkungan Linux, kepopuleran ini karena ditunjang karena performansi *query* dari *database*-nya yang saat itu bisa dikatakan paling cepat dan jarang bermasalah (Buana, 2014).

2.2.6 XAMPP

Menurut Buana (2014), “XAMPP adalah perangkat lunak *opensource* yang diunggah secara gratis dan bisa dijalankan di semua operasi seperti *windows, linux, solaris, dan mac*”. Tampilan panel XAMPP dapat di tunjukan pada Gambar 2.1. Gambar 2.2.



Gambar 2.1 Tampilan panel XAMPP

2.2.7 Sublime

Menurut (Faridl, 2015), Sublime text adalah teks editor berbasis Python, sebuah teks editor yang elegan, kaya fitur, cross platform, mudah dan simpel yang cukup terkenal di kalangan developer (pengembang), penulis dan desainer. Para programmer biasanya menggunakan sublime text untuk menyunting source code yang sedang ia kerjakan. Sampai saat ini sublime text sudah mencapai versi 3. Sublime text mempunyai beberapa keunggulan-keunggulan yang dapat membantu pengguna dalam membuat sebuah *web development*. contoh *script* menggunakan *software* sublime dapat di tunjukan pada Gambar 2.2.



Gambar 2.2 *Script* menggunakan *software* sublime.

2.2.8 *Website*

Menurut Kadir (2015), *Website* merupakan kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara, dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait, yang masing- masing masing dihubungkan dengan jaringan-jaringan halaman.=

2.2.9 **Definisi UML**

Menurut Rosa dan Shalahuddin (2015), dijelaskan 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.

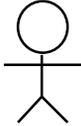
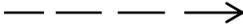
Pada perkembangan teknik pemrograman berorientasi objek, muncul sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu *Unified Modelling Language* (UML).

Untuk membangun sistem pakar perawatan sepatu peneliti menggunakan 4 macam diagram *Unified Modeling Language* (UML), yaitu :

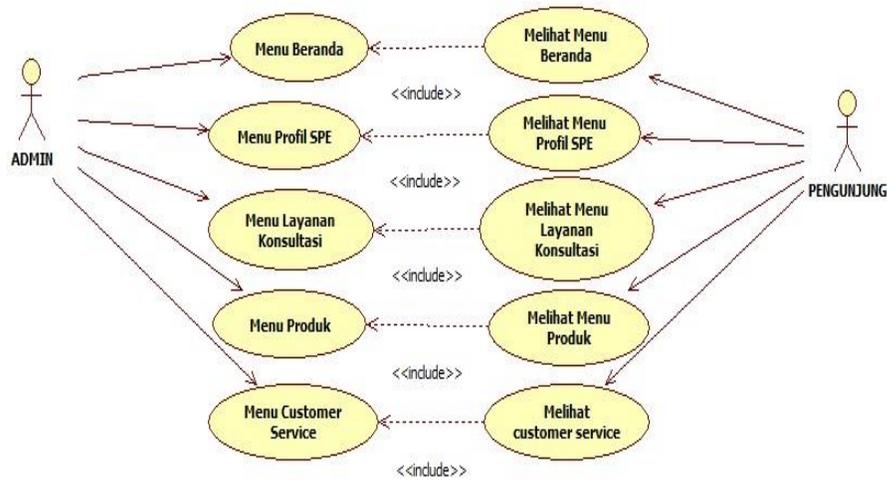
1. *Use Case Diagram*

Use case atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol *use case diagram* dapat dilihat pada Tabel 2.1.

Tabel 2.1 Simbol *Use Case*

No.	Simbol	Nama	Deskripsi
1		<i>Actor</i>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri
2		<i>Use case</i>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor
3		<i>Association</i>	Komunikasi antara aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor
4	<p><<extend>></p> 	<i>Extend</i>	Relasi use case tambahan ke sebuah use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan itu
5		<i>Generalization</i>	Menghubungkan generalisasi dan spesialisasi (umum-khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari
6	<p><<include>></p> 	<i>Include</i>	Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya atau sebagai syarat dijalankan use case ini

Rosa dan Shalahuddin (2015)



Gambar 2.3 Contoh use case diagram

2.2.10 Activity Diagram

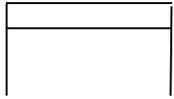
Rosa dan Shalahuddin (2015) “*activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak”.

Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

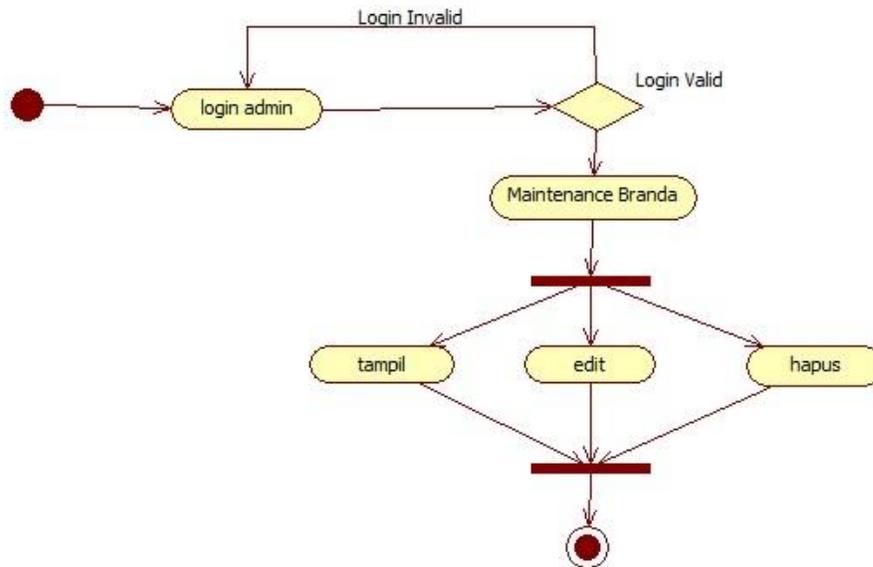
Adapun simbol-simbol yang digunakan dalam activity diagram di tabel 2.2.

Tabel 2.2 Simbol Activity Diagram.

No	Simbol	Nama	Deskripsi
1	●	Status awal	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
2	▭	Aktivitas	Aktivitas yang dilakukan sistem, biasanya diawali dengan kata kerja
3	◇	Decision	Asosiasi percabangan dimana jika adapilihan aktivitas lebih dari satu

No	Simbol	Nama	Deskripsi
4		<i>Join</i>	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
5		Status akhir	Status akhir yang dilakukan sebuah sistem, sebuah diagram aktivitas memiliki sebuah status akhir
6		<i>Swimlane</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

Rosa dan Shalahuddin (2015)



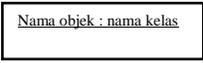
Gambar 2.4 Contoh *activity diagram*

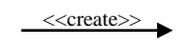
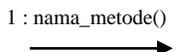
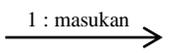
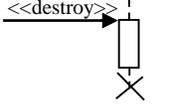
2.2.11 *Sequence Diagram*

Rosa dan Shalahuddin (2015), “diagram sekuen menggambarkan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek”.

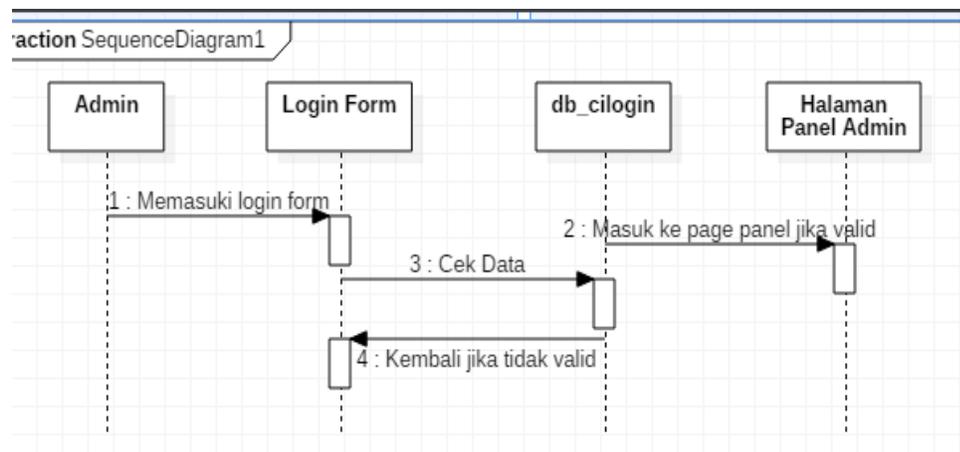
Sequence diagram menunjukkan urutan event kejadian dalam suatu waktu. Komponen *sequence* diagram terdiri atas obyek yang dituliskan dengan kotak segiempat bernama message diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan progress vertikal. Simbol-simbol yang digunakan dalam sequence dapat ditunjukkan pada Tabel 2.3.

Tabel 2.3 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.

NO	SIMBOL	NAMA	KETERANGAN
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 destroy	Menyatakan suatu objek mengakhiri hidup objek lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada destroy.

Rosa dan Shalahuddin (2015)



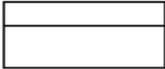
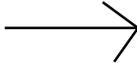
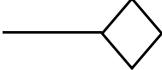
Gambar 2.5 Contoh *Sequence diagram*

2.2.12 *Class Diagram*

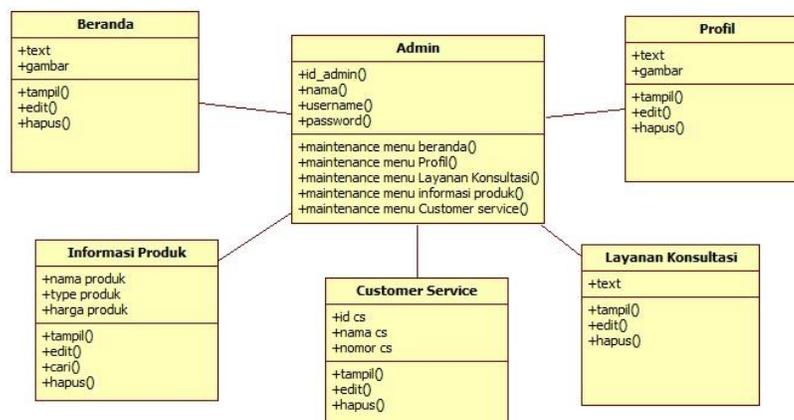
Rosa dan Shalahuddin (2015), “*classdiagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem”.

Diagram kelas dibuat agar pembuat program atau *programmer* membuat kelas-kelas sesuai rancangannya dalam diagram kelas agar antara dokumentasi perancangan perangkat lunak sinkron. Adapun simbol-simbol yang digunakan dalam *classdiagram* dapat di tunjukan pada Tabel 2.4.

Tabel 2.4 Simbol *Class Diagram*

No	Gambar	Nama	Deskripsi
1		<i>Class</i>	Kelas pada sistem
2		<i>Interface</i>	Sama dengan konsep <i>interface</i> 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
5		<i>Generalization</i>	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
6		<i>Dependency</i>	Relasi antar kelas dengan makna ketergantungan antar kelas
7		<i>Aggregation</i>	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>)

Rosa dan Shalahuddin (2015)

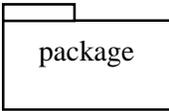
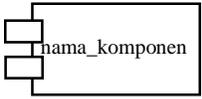


Gambar 2.6 Contoh *class diagram*

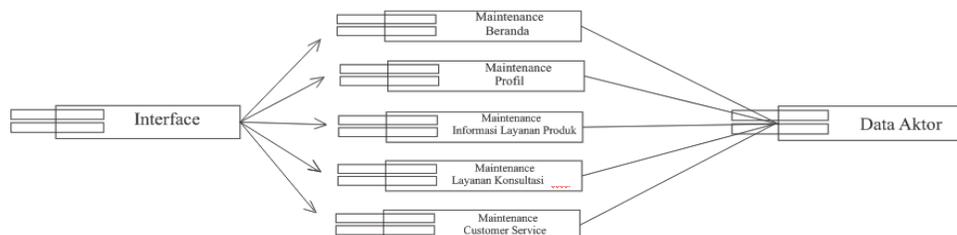
2.2.13 Component Diagram

Rosa dan Shalahuddin (2015) Diagram komponen atau *component diagram* dibuat untuk menunjukkan organisasi dan ketergantungan diantara kumpulan komponen dalam sebuah sistem. *Component diagram* fokus pada komponen sistem yang dibutuhkan dan ada didalam sistem. *Component diagram* juga dapat digunakan untuk memodelkan *source code* program perangkat lunak, komponen *executable* yang dilepas ke *user*, basis data secara fisik, sistem yang harus beradaptasi dengan sistem lain dan *framework* sistem. Simbol *component diagram* dapat dilihat pada Tabel 2.5.

Tabel 2.5 Simbol *Component Diagram*.

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Package</i>	<i>Package</i> merupakan sebuah bungkus 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

Rosa dan Shalahuddin (2015)

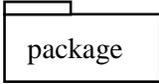
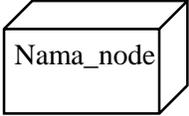


Gambar 2.7 Contoh *component diagram*

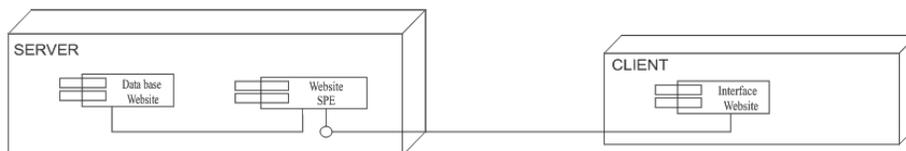
2.2.14 Deployment Diagram

Rosa dan Shalahuddin (2015) *deployment* diagram menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi. Diagram deployment juga dapat digunakan untuk memodelkan sistem tambahan yang menggambarkan rancangan device, node, dan hardware. Sistem client, sistem terdistribusi murni, rekayasa ulang aplikasi. Simbol *deployment* diagram ditunjukkan pada Tabel 2.6.

Tabel 2.6 Simbol *Deployment Diagram*

NO	SIMBOL	NAMA	KETERANGAN
1.		<i>Package</i>	<i>Package</i> merupakan sebuah bungkus 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> .

Rosa dan Shalahuddin (2015)



Gambar 2.8 Contoh *component diagram*

2.2.15 Bootstrap

Alatas (2013:2), Bootstrap merupakan framework ataupun tools untuk membuat aplikasi web ataupun situs web responsive secara tepat, mudah dan gratis.

Bootstrap merupakan framework HTML, CSS, Javascript populer untuk membangun situs web yang responsive. RWD (Responsive Web Design) adalah desain situs yang otomatis akan menyesuaikan diri agar tampil baik di semua perangkat dari ponsel sampai dekstop. Bootstrap terdiri dari satu set file css, javascript, dan jquery.

File Bootstrap

Satu set file bootstrap dapat diperoleh di situs getbootstrap.com yang terdiri dari:

- a) direktori css: bootstrap.css
- b) direktori js: bootstrap.js
- c) direktori font: file-file font glyphicons
- d) 1 file jquery.js yang dapat diperoleh di jquery.com.

2.2.16 MVC

Pola pemrograman Model View Controller (MVC) merupakan pola pemrograman yang memisahkan aplikasi kedalam 3 bagian yang membuat program logic dan presentation logic tidak tercampur aduk menjadi satu. MVC dikembangkan pada tahun 1970-an oleh Prof. Trygve Reenskaug dari Smalltalk (Deacon, 2009 : 1).

2.2.17 Definisi *Black Box Testing*

Black Box Testing yaitu untuk memenuhi *unit* program apakah sudah sesuai kebutuhan (*requirement*) yang digunakan dalam spesifikasi. Cara pengujian hanya dilakukan dengan menjalankan atau mengeksekusi *unit*, kemudian diamati apakah hasil dari *unit* sesuai dengan proses yang diinginkan (Al Fattah, 2007).

Menurut Rosa dan Shalahuddin (2015) *Black Box Testing* (Pengujian Kotak Hitam) yaitu "menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program".

2.2.18 Definisi Teknik Linkert

Skala Likert merupakan metode skala bipolar yang mengukur baik tanggapan positif ataupun negatif terhadap suatu pernyataan. Empat skala pilihan juga kadang

digunakan untuk kuesioner skala Likert yang memaksa orang memilih salah satu kutub karena pilihan “netral” tak tersedia. (Nasir N, 2005).