

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Hasil Penelitian sebelumnya yang menjadi kajian untuk menyelesaikan masalah yang dihadapi dan melakukan pengkajian dari teori dan metode pemecahan masalah yang sama dari permasalahan yang ada meliputi penelitian-penelitian dibawah ini.

Dalam penelitian yang dilakukan Diana Nur Asniati (2016) dalam skripsinya yang berjudul Sistem Pengolahan Data Bengkel Ban dan Sparepart Pada Distro Ban. Penelitian ini membahas permasalahan dalam pengolahan data di toko Distro Ban yang masih dilakukan dengan cara manual yaitu dengan cara menuliskan data ke buku, sehingga kemungkinan terjadi kesalahan dalam perhitungan maupun ketersediaan stok barang yang ada menjadi tidak akurat dan tidak terkontrol. Dari permasalahan tersebut dibangunlah sebuah sistem aplikasi pengolahan data bengkel dan *sparepart* pada toko Distro Ban supaya pengolahan data dapat dilakukan dengan akurat dan efektif. Sistem ini digunakan untuk mengolah data administrasi meliputi proses pelayanan pelanggan yang melakukan pembelian *sparepart* atau yang melakukan servis di toko Distro Ban. Dalam laporan ini menghasilkan sistem pengolahan data bengkel dan *sparepart* pada toko Distro Ban berbasis komputer secara sistematis, tepat, cepat dan akurat. sistem juga berguna dalam pencarian data dan pembuatan laporan pelayanan servis dan penjualan *sparepart* yang lebih teoat dan akurat.

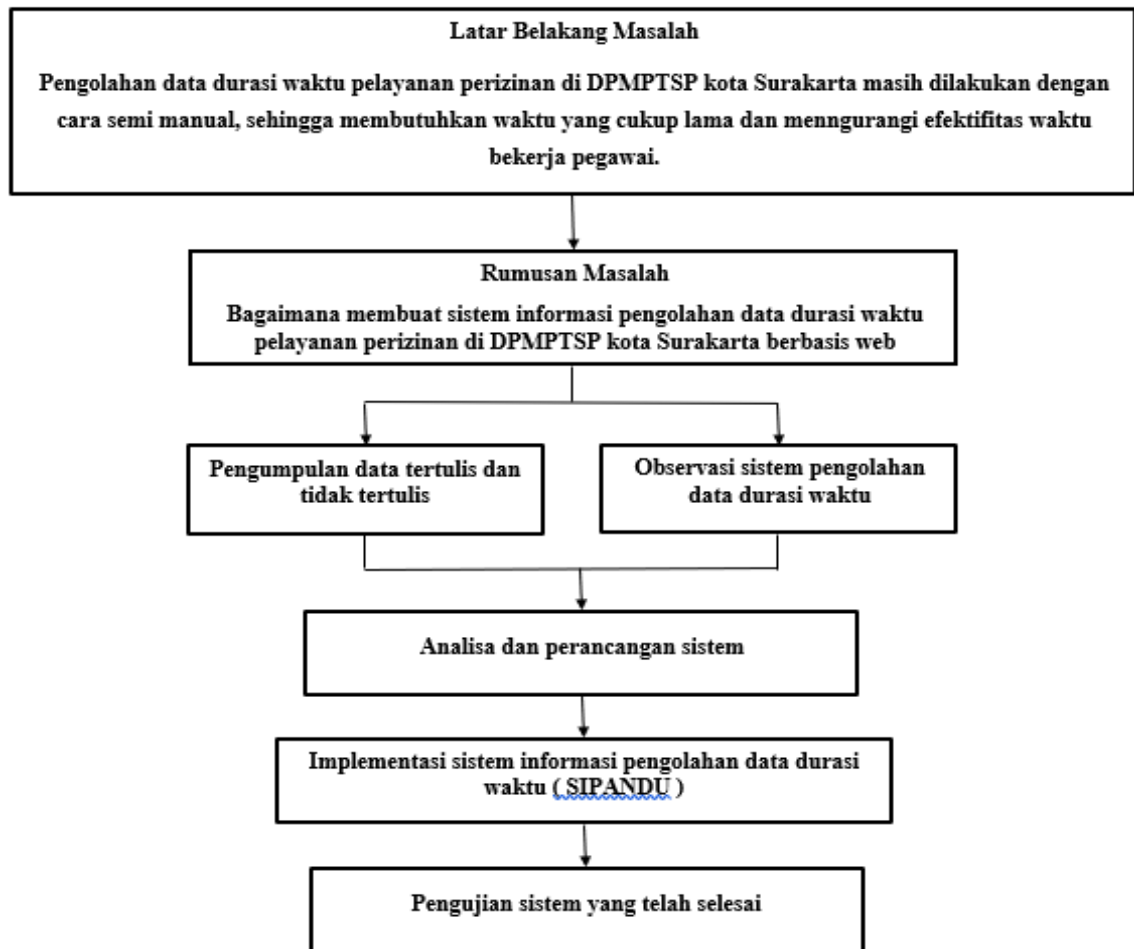
Dalam penelitian yang dilakukan Indo Pratono (2016) dalam skripsinya yang berjudul Aplikasi Pengolahan Nilai Rapor Siswa SMP Negeri 1 Sawit Berbasis Web. Penelitian ini membahas permasalahan dalam pengolahan nilai rapor siswa yang masih dilakukan secara konvensional yaitu dengan aplikasi microsoft excel. Dari pengolahan yang masih dilakukan dengan cara konvensional tersebut menimbulkan beberapa masalah, yaitu membutuhkan waktu yang lama dalam proses perhitungan nilai, masih terdapat beberapa kesalahan perhitungan nilai rata-rata siswa, proses pengolahan nilai ini dilakukan oleh guru mata pelajaran sehingga

sering menyebabkan keterlambatan dalam penyusunan nilai rapor, pada setiap akhir semester guru disibukkan merekap nilai rapor. Dari permasalahan tersebut dibangunlah sebuah sistem pengolahan nilai rapor siswa di SMP Negeri Sawit supaya pengolahan nilai rapor dapat dilakukan dengan cepat, efektif, dan akurat. sistem pengolahan nilai rapor ini terdiri dari beberapa pengolahan data master, yaitu data siswa, data admin, data guru, data kelas, data ekstrakurikuler, dan data mata pelajaran. Dalam penelitian ini dihasilkan sistem

Dalam penelitian yang dilakukan Angky Sudarmono (2015) yang berjudul Sistem Komputerisasi dan Pengolahan Data di Puskesmas Kecamatan Teras, Kabupaten Boyolali. Penelitian ini membahas permasalahan pada proses pengolahan data di Puskesmas Kecamatan Teras, kabupaten Boyolali yang masih dilakukan dengan cara manual, misalnya dalam proses pendaftaran masih memanfaatkan data berupa *hardcopy* dengan media kertas. lalu dalam pendaftaran pasien apabila kartu berobat pasien tertinggal atau hilang, maka petugas harus membuat kartu baru sehingga mengakibatkan penduplikasian data. Pencarian data rekam medik pasien juga membutuhkan waktu yang lama karena data masih disimpan dalam satu rak. Tujuan dari penelitian ini adalah membuat suatu sistem yang akan meminimalkan kesalahan yang timbul akibat pencatatan aktivitas yang masih dilakukan dengan manual, sehingga dapat menunjang kelancaran dan juga meningkatkan kualitas di puskesmas Teras dalam mengolah data. Dalam penelitian ini menghasilkan sebuah sistem pengolahan data puskesmas kecamatan Teras berbasis komputerisasi sehingga proses pengolahan data di puskesmas secara cepat, tepat dan akurat sehingga dapat memperkecil kesalahan-kesalahan dalam pengarsipan.

2.2 Kerangka Pemikiran

Berikut ini adalah kerangka pemikiran yang digunakan di dalam melakukan analisis dan perancangan sistem informasi pengolahan data durasi waktu pelayanan perizinan di dinas penanaman modal dan pelayanan terpadu kota Surakarta yang ditunjukkan pada Gambar 2.1.



Gambar 2.1 Kerangka Pemikiran

Penjelasan dari kerangka pemikiran tersebut adalah :

1. Latar Belakang Masalah

Pengolahan data durasi waktu pelayanan perizinan di Dinas Penanaman Modal dan Pelayanan Terpadu Satu Pintu Kota Surakarta masih menggunakan cara semi manual. Proses ini membutuhkan waktu yang lama karena admin harus mengekspor data dari *database* SIPINTER menggunakan Navicat dan mengolahnya di Microsoft Excel sehingga mengurangi efektifitas waktu bekerja pegawai.

2. Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan maka rumusan masalahnya adalah bagaimana membuat sistem informasi pengolahan data durasi waktu pelayanan perizinan di DPMPTSP kota Surakarta berbasis *web*.

3. Pengumpulan data tertulis dan tidak tertulis

Tahap pengumpulan data pada penelitian ini melalui observasi, wawancara dan studi literatur. Pengumpulan data bertujuan untuk mengetahui permasalahan dan kebutuhan informasi mengenai pengolahan data durasi waktu di DPMPTSP.

4. Observasi Sistem Pengolahan Data Durasi Waktu

Adalah tahap pengamatan contoh sistem yang telah ada, buku, jurnal, dan karya ilmiah yang dapat dijadikan sebagai referensi untuk pembuatan sistem.

5. Analisis sistem dan perancangan sistem secara terstruktur.

6. Implementasi sistem informasi pengolahan data durasi waktu (SIPANDU)

Implementasi sistem adalah tahap penerapan dari sistem yang telah melewati proses analisa dan perancangan sistem. Tahapan ini adalah tahap lanjutan dari sistem yang telah disetujui dari lingkup desain dan juga *coding* dalam proses perancangan. Proses ini untuk mengetahui apakah sistem telah berjalan dan siap digunakan.

7. Pengujian Sistem

Pengujian sistem merupakan tahap setelah sistem berhasil diimplementasi. Pengujian sistem bertujuan untuk mengetahui kelebihan, kelemahan serta mengetahui kelayakan suatu sistem untuk digunakan.

2.3 Teori Pendukung

2.3.1 Sistem Informasi

Menurut Kadir (2014), Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi bersifat menajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan laporan-laporan yang diperlukan oleh pihak luar tertentu dengan informasi yang diperlukan untuk pengambilan suatu keputusan.

2.3.2 Website

Menurut Nugroho dalam Aprisa (2015), *Website* atau situs dapat diartikan sebagai kumpulan halaman-halaman yang berasal dari *file-file* berisi bahasa

pemrograman yang saling berhubungan digunakan untuk menampilkan informasi, gambar bergerak dan tidak bergerak, suara dan atau gabungan dari semuanya itu baik yang bersifat statis maupun dinamis.

2.3.3 *Framework*

Dalam penelitian Thwari dan Yedey (2016), Mengatakan *framework* adalah seperangkat perpustakaan yang disusun dalam desain arsitektural untuk memberikan kecepatan, akurasi. *Framework* adalah kerangka kerja, *framework* juga dapat di artikan sebagai kumpulan *script* (terutama *class* dan *function* yang dapat membantu *developer* atau *programmer* dalam menagani berbagai masalah-masalah dalam pemrograman seperti koneksi ke *database*, pemanggilan variabel, dan file. Sehingga *developer* lebih fokus dan lebih cepat membangun aplikasi.

2.3.4 *Codeigniter*

Dalam penelitian Hustinawati dkk (2014), Mengatakan *codeIgniter* adalah *framework* aplikasi *web* yang bersifat *open source* yang digunakan untuk membangun aplikasi PHP yang dinamis. Tujuan utamanya adalah untuk membantu pengembangan pengembang *codeIgniter* untuk mengerjakan aplikasi lebih cepat daripada menulis semua kode dari nol.

2.3.5 *Data*

Menurut Gordon B. Davis dalam buku karya Hutahaean (2014), Data adalah bahan mentah bagi informasi, dirumuskan sebagai kelompok lambang-lambang tidak acak menunjukkan jumlah-jumlah, tindakantindakan, hal-hal dan sebagainya.

2.3.6 *MySQL (My Structure Query Language)*

Menurut Madcoms (2016), MySQL adalah sistem manajemen *database* SQL yang bersifat *open source* dan paling populer saat ini. Sistem *database* MySQL mendukung beberapa fitur seperti *multithreaded*, *multi-user*, dan SQL *database management system* (DBMS). *Database* ini dibuat untuk keperluan sistem *database* yang cepat, handal, dan mudah digunakan

2.3.7 Pengolahan Data

Pengertian pengolahan data menurut George R. Terry dalam buku karya Hutahaean (2014), Pengolahan data adalah serangkaian operasi atas informasi yang direncanakan guna mencapai tujuan atau hasil yang diinginkan.

2.3.8 *PHP Hypertext Preprocessor (PHP)*

Menurut Hakim (2014), PHP merupakan bahasa pemrograman yang bisa digunakan untuk membuat halaman HTML. File .php yang dibuat akan diproses di dalam *server*, sedangkan halaman yang akan dikirim ke *browser* pengunjung hanyalah tampilan HTML-nya. Dengan PHP, halaman *website* yang dibuat akan menjadi dinamis, yakni dapat selalu berubah tanpa harus mengubah isi *website* secara manual. Informasi akan diproses ulang oleh *web server* sehingga akan didapatkan isi paling mutakhir dari halaman *web*.

2.3.9 HTML

Menurut Sandi (2014), HTML atau Hyper Text Markup Language adalah sebuah bahasa markah yang digunakan untuk membuat sebuah halaman *web*, menampilkan berbagai informasi di dalam sebuah penjelajah *web* internet dan *formatting hypertext* sederhana yang ditulis ke dalam berkas format ASCII agar dapat menghasilkan tampilan wujud yang terintegrasi. HTML adalah standar yang digunakan secara luas untuk menampilkan halaman *web*. HTML saat ini merupakan standar internet yang didefinisikan dan dikendalikan penggunaannya oleh World Wide Web Consortium (W3C).

2.3.10 Xampp

Menurut Madcoms (2016), Xampp adalah sebuah paket kumpulan *software* yang terdiri dari Apache, MySQL, phpMyAdmin, PHP, Perl, Filezilla dan lain-lain yang berfungsi untuk memudahkan instalasi lingkungan PHP, dimana biasanya lingkungan pengembangan *web* memerlukan PHP, Apache, MySQL, dan phpMyAdmin serta *software* lainnya yang terkait dengan pengembangan *web*.

2.3.11 CSS

Menurut Hartono (2015) CSS (*Cascading Style Sheet*) adalah kumpulan aturan-aturan pemformatan yang mengontrol tampilan konten dalam sebuah halaman *web*. Terdapat tiga jenis CSS, yaitu:

- a. *Inline style sheet* : cukup menambahkan atribut *style* di *tag* yang ingin kita berikan pemformatan.
- b. *Internal style sheet* : meletakkan aturan pemformatan dengan CSS dibagi `<head>` dari *html* dengan tambahan *tag* `<style>`.
- c. *External style sheet* : memisahkan antara file CSS dengan file HTML-nya.

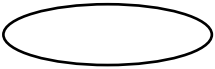
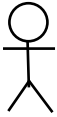

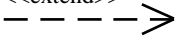
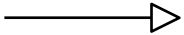

2.3.12 DPMPTSP Kota Surakarta

Dinas Penanaman Modal dan Pelayanan Terpadu Satu Pintu kota Surakarta, atau lebih dikenal dengan DPMPTSP kota Surakarta adalah lembaga pemerintahan di lingkup pemerintah kota Surakarta yang melayani perizinan dan penanaman modal di wilayah kota Surakarta, meliputi perizinan dan *non*-perijinan yang prima dan satu pintu dengan kemudahan-kemudahan, biaya yang transparan, serta ketepatan waktu penyelesaian.

2.3.13 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 (Sukamto & Shalahuddin (2016)). Simbol *use case diagram* dapat dilihat pada Tabel 2.1.

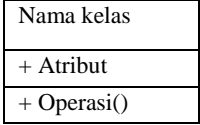


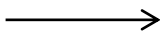
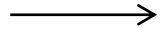
Tabel 2.1 Simbol-simbol *Use Case Diagram* (Sukamto & Shalahuddin (2016))

| NO | SIMBOL | NAMA | KETERANGAN |
|----|--|-----------------------|---|
| 1 | <p>Nama <i>Use Case</i></p>  | <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 | <p><<extend>></p>  | <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>Generalization</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 | <p><<include>></p>  | <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. |

2.3.14 Class Diagram

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode operasi. Kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem sehingga pembuat perangkat lunak dapat membuat kelas-kelas di dalam program perangkat lunak sesuai dengan perancangan diagram kelas (Sukamto & Shalahuddin (2016)). Simbol *class diagram* dapat dilihat pada Tabel 2.2.



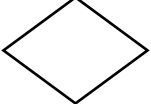
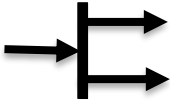

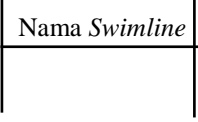

Tabel 2.2 Simbol-simbol *Class Diagram* (Sukamto & Shalahuddin (2016))

| 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>). |

2.3.15 Activity Diagram

Menurut Widodo, Prabowo P dan Herlawati (2011) *Activity diagram* adalah teknik untuk menggambarkan logika prosedural, proses bisnis, dan jalur kerja. Dalam beberapa hal, diagram ini memainkan peran mirip sebuah diagram alir, tetapi perbedaan prinsip antara diagram ini dengan notasi diagram alir adalah diagram ini mendukung *behavior parallel*. Simbol *Activity Diagram* ditunjukkan pada Tabel 2.3.



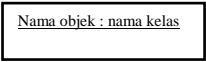

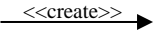
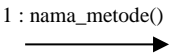
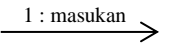
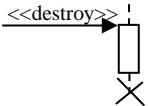
Tabel 2.3 Simbol-simbol *Activity Diagram* (Widodo, dan Herlawati (2011))

| 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 jika ada pilihan aktivitas lebih dari satu. |
| 4 |  | <i>Fork</i> | Asosiasi percabangan jika ada aktivitas dicabangkan menjadi dua atau lebih. |
| 5 |  | <i>Join</i> | Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu. |
| 6 |  | <i>Swimlane</i> | Cara mengelompokkan aktivitas berdasar aktor |
| 7 |  | Status akhir | Status akhir yang dilakukan sistem. |

2.3.16 Sequence Diagram

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek (Sukamto & Shalahuddin (2016)). *Sequence* diagram dapat dilihat pada Tabel 2.4.

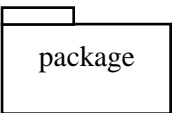
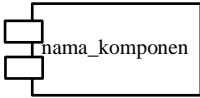



Tabel 2.4 Tabel *Sequence* Diagram (Sukamto & Shalahuddin (2016))

| 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 dikirimi. |
| 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> . |

2.3.17 Component Diagram

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 (Sukamto & Shalahuddin (2016)). Simbol *component* diagram dapat dilihat pada Tabel 2.5.

Tabel 2.5 Simbol *Component* Diagram. (Sukamto & Shalahuddin (2016))

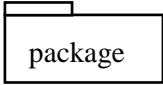
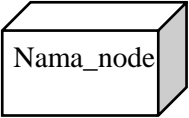
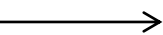

| NO | GAMBAR | NAMA | KETERANGAN |
|----|---|-------------------|--|
| 1 |  | <i>Package</i> | <i>Package</i> merupakan sebuah bungkusuan dari satu atau lebih komponen |
| 2 |  | <i>Component</i> | Komponen sistem |
| 3 |  | <i>Dependency</i> | Kebergantungan antar komponen, arah panah mengarah pada 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 |

2.3.18 Deployment Diagram

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 (Sukamto & Shalahuddin (2016)). Simbol *deployment* diagram ditunjukkan pada Tabel 2.6

Tabel 2.6 Simbol - Simbol Deployment (Sukamto & Shalahuddin (2016))

| 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> . |

2.4 Pengujian *WebQual*

Menurut Sukamto & Shalahuddin (2016), Pengujian perangkat lunak adalah sebuah elemen topik yang memiliki cakupan luas dan sering dikaitkan dengan verifikasi (*verification*) dan validasi (*validation*). Verifikasi mengacu pada sekumpulan aktifitas yang menjamin bahwa perangkat lunak mengimplementasikan dengan benar sebuah fungsi yang spesifik. Validasi mengacu pada sekumpulan aktifitas yang berbeda yang menjamin bahwa perangkat lunak yang dibangun dapat ditelusuri sesuai dengan kebutuhan pelanggan (*customer*). Penelitian ini menggunakan model pengujian *WebQual*.

Menurut Iman Sanjaya dalam jurnal Penelitian IPTEK-KOM, *WebQual* merupakan salah satu metode atau teknik pengukuran kualitas *website* berdasarkan persepsi pengguna akhir. Metode ini merupakan pengembangan dari SERVQUAL yang banyak digunakan sebelumnya pada pengukuran kualitas jasa. *WebQual* sudah mulai dikembangkan sejak tahun 1998 dan telah mengalami beberapa interaksi dalam penyusunan dimensi dan butir pertanyaannya.

Menurut teori *WebQual* pada jurnal yang disusun oleh Anwariningsih (2011) terdapat tiga dimensi yang mewakili kualitas suatu *website*, yaitu kegunaan (*usability*), kualitas informasi (*information quality*) dan interaksi layanan (*service interaction*). Masing-masing dimensi terdiri dari beberapa pernyataan yang ditunjukkan pada Tabel 2.7 sampai Tabel 2.9.

Tabel 2.7 Dimensi Kegunaan (*usability*)

| No | Deskripsi Indikator |
|----|--|
| 1 | <i>Website</i> mudah untuk dioperasikan. |
| 2 | Interaksi dengan <i>website</i> sangat mudah dimengerti dan tidak membingungkan. |
| 3 | Pengguna merasa mudah untuk bernavigasi dalam <i>website</i> . |
| 4 | <i>Website</i> mudah untuk digunakan. |
| 5 | <i>Website</i> memiliki tampilan yang menarik. |
| 6 | Desain <i>website</i> sesuai dengan tipe <i>website</i> . |
| 7 | <i>Website</i> menunjukkan kemampuannya. |
| 8 | <i>Website</i> dapat memberikan pengaruh/ pengalaman positif bagi pengguna. |

Tabel 2.8 Dimensi Kualitas Informasi (*information quality*)

| No | Deskripsi Indikator |
|----|---|
| 1 | <i>Website</i> menyajikan informasi yang akurat. |
| 2 | Informasi yang disajikan <i>website</i> dapat dipercaya. |
| 3 | Informasi yang disajikan tepat waktu atau <i>up to date</i> . |
| 4 | Informasi yang disajikan relevan. |
| 5 | Informasi yang disajikan mudah dipahami. |
| 6 | Informasi yang disajikan sangat detail. |
| 7 | Informasi disajikan dalam format yang sesuai. |

Tabel 2.9 Dimensi Kualitas Interaksi (*interaction quality*)

| No | Deskripsi Instansi |
|----|--|
| 1 | <i>Website</i> memiliki reputasi yang baik. |
| 2 | Pengguna merasa aman untuk melakukan transaksi atau interaksi dengan <i>website</i> . |
| 3 | <i>Website</i> menjaga informasi pribadi pengguna. |
| 4 | <i>Website</i> memberi ruang untuk personalisasi. |
| 5 | <i>Website</i> memberi ruang untuk komunitas. |
| 6 | <i>Website</i> memudahkan pengguna untuk berkomunikasi dengan organisasi. |
| 7 | Pengguna merasa yakin dengan layanan/informasi yang disediakan karena sesuai dengan yang dijanjikan. |

2.5 Uji Validitas

Menurut (Sugiyono, 2010) valid berarti instrumen tersebut dapat digunakan untuk mengukur apa yang hendak diukur. Suatu kuesioner dinyatakan valid apabila r hitung $>$ r tabel, sebaliknya kuesioner dapat dinyatakan tidak valid apabila r hitung $<$ r tabel. Pengujian validitas ini menggunakan batasan r tabel dengan signifikansi 0,05 dan uji 2 sisi.

2.6 Uji F (Uji Simultan)

Menurut (Ghozali, 2011) uji F digunakan untuk menunjukkan apakah semua variabel independent yang dimasukkan dalam model mempunyai pengaruh secara bersama-sama terhadap variabel dependent. Dasar penerimaan atau penolakan hipotesis dapat dilihat dengan membandingkan F hitung dengan F tabel, jika F hitung $>$ F tabel maka H_0 ditolak dan H_a diterima.

2.7 Uji T (Uji Parsial)

Menurut (Ghozali, 2011), uji T pada dasarnya menunjukkan seberapa jauh pengaruh satu variabel independent secara individual dalam menerangkan variabel

dependent. Penerimaan atau penolakan hipotesis dilakukan dengan kriteria sebagai berikut:

1. Jika nilai $sig < 0,05$ atau $t \text{ hitung} > t \text{ tabel}$ maka terdapat pengaruh variabel X (variabel bebas) terhadap variabel Y (variabel terikat).
2. Jika nilai $sig > 0,05$ atau $t \text{ hitung} < t \text{ tabel}$ maka tidak terdapat pengaruh variabel X (variabel bebas) terhadap variabel Y (variabel terikat).