

BAB II

LANDASAN TEORI

1.1 STIKOM Career Center (SCC)

STIKOM Career Center (SCC) merupakan sebuah departemen yang ada di STIKOM Surabaya. SCC menjembatani antara mahasiswa STIKOM Surabaya dengan perusahaan/instansi yang menyediakan lapangan pekerjaan. Selain itu, SCC juga memberikan pembekalan dan pelatihan kepada mahasiswa yang akan memasuki dunia kerja (SCC, 2012).

Visi SCC adalah sebagai pusat informasi kerja dan pengembangan bagi mahasiswa dan alumni STIKOM Surabaya agar mampu bersaing di dunia kerja.

Misi SCC antara lain, yaitu:

1. Membangun kemitraan strategis antara STIKOM Surabaya dengan perusahaan/institusi/industri.
2. Menjembatani kebutuhan mahasiswa dan alumni STIKOM Surabaya dengan perusahaan/institusi/industri termasuk dengan instansi pemerintah.
3. Menyiapkan calon-calon sarjana untuk memiliki kemampuan, keahlian, keterampilan, dan pengetahuan dalam hal pengembangan diri.

1.2 Komunitas

Komunitas adalah jaringan dari beberapa individu yang saling mengikat dan meningkatkan sosialisasi sesama jaringan, saling mendukung, memberikan informasi, adanya rasa memiliki dan menjadi identitas sosial (Delanty, 2003). Ikatan yang kuat dan dukungan dari sesama anggota komunitas memungkinkan adanya saling ketergantungan di antara anggota komunitas yang secara sadar atau

tidak terjadi interaksi saling memanfaatkan di antara anggota komunitas. Perasaan saling berbagi dibutuhkan dalam komunitas di mana anggota akan bersimpati terhadap ide-ide yang ada di sekitar di mana kelompok tersebut berasal. Komunitas dapat dikatakan sebagai sekelompok orang yang saling mempengaruhi, memiliki kesamaan identitas kelompok dan memiliki ikatan emosional antar anggotanya.

1.2.1 Ciri-Ciri Komunitas

Suatu komunitas memiliki 4 (empat) ciri utama:

1. Adanya keanggotaan di dalamnya, tidak mungkin ada komunitas tanpa ada anggota.
2. Adanya saling mempengaruhi, anggota komunitas bisa saling mempengaruhi satu dengan yang lainnya.
3. Adanya integrasi dan pemenuhan kebutuhan antar anggota.
4. Adanya ikatan emosional antar anggota.

1.2.2 Manfaat Komunitas

Manfaat komunitas antara lain adalah (Delanty, 2003):

1. Menampung ide-ide yang berasal dari masyarakat luas, sehingga dapat dipilih ide yang tepat untuk dijadikan kebijakan bagi masyarakat sosial.
2. Mengungkapkan ikatan-ikatan dalam masyarakat umum dan juga sosialisasinya.
3. Mengungkapkan relasi sosial secara spesifik, dalam hubungan dengan negara yang bersifat autokratik.

4. Menghubungkan arti dunia dengan segera, dan berkait dengan kehidupan sehari-hari.

1.3 Alumni

Alumni adalah orang-orang yang telah mengikuti atau tamat dari suatu sekolah atau perguruan tinggi (Almanfaluthi, 2009). Alumni merupakan bagian yang tidak dapat dipisahkan dalam sebuah siklus pendidikan. Alumni menjadi penghubung kampus dengan dunia global. Alumni juga berfungsi sebagai media yang menyampaikan visi dunia kepada kampus.

1.4 *Tracer Study*

Penelusuran lulusan (*Tracer Study*) merupakan bagian penting dari aktivitas sebuah lembaga pendidikan (Wuradji, 2010). Dengan adanya kegiatan *tracer study*, memungkinkan sebuah lembaga pendidikan melacak kondisi lulusan yang telah dihasilkan. Kemudian dari informasi yang diperoleh, dapat diambil berbagai kebijakan dan tindakan yang manfaat bagi para lulusan dan bagi pengembangan lembaga tersebut.

Penelusuran lulusan juga dapat digunakan untuk mengetahui kinerja dan produktivitas lulusan setelah mereka terjun ke masyarakat. *tracer study* juga dapat menjawab pertanyaan apakah pengalaman yang diperoleh selama mengikuti proses pembelajaran, turut menentukan perkembangan karir di masa mendatang. Dengan kata lain, apakah ada relevansi antara pendidikan yang telah diterima oleh lulusan dengan karir mereka.

Indikator yang jelas tentang jumlah, profil kerja masa mendatang, pelatihan yang diperlukan oleh lembaga pendidikan dapat diperoleh dari

pelaksanaan *tracer study*. Badan Akreditasi Nasional Perguruan Tinggi (BAN-PT) dalam sistem akreditasinya juga mengatur tentang studi pelacakan alumni. Aspek lulusan merupakan salah satu dari tujuh standar akreditasi yang telah ditetapkan oleh BAN-PT. Elemen penilaian yang berkaitan dengan lulusan pada standar akreditasi ke-3 (mahasiswa dan lulusan) untuk proses akreditasi mencakup hal-hal berikut (BAN-PT, 2008):

1. Kebijakan sistem rekrutmen dan seleksi calon mahasiswa (mencakup mutu prestasi dan reputasi akademik serta bakat pada jenjang pendidikan sebelumnya, equitas wilayah, kemampuan ekonomi dan gender) dan pengelolaan lulusan dan alumni (mencakup layanan alumni, peran dalam asosiasi profesi atau bidang ilmu, dukungan timbal balik alumni).
2. Profil lulusan: ketepatan waktu penyelesaian studi, proporsi mahasiswa yang menyelesaikan studi dalam batas masa studi.
3. Layanan dan pendayagunaan lulusan: ragam, jenis, wadah, mutu, harga, identitas.
4. Pelacakan dan perekaman data lulusan: kekomprehensifan, pemutakhiran, profil masa tunggu kerja pertama, kesesuaian bidang kerja dengan bidang studi, dan posisi kerja pertama.
5. Partisipasi lulusan dan alumni dalam mendukung pengembangan akademik dan non-akademik program studi.

Sebagai salah satu keluaran langsung dari proses pendidikan yang dilakukan oleh program studi sarjana, lulusan yang bermutu memiliki ciri penguasaan kompetensi akademik termasuk *hardskills* dan *softskills* sebagaimana

dinyatakan dalam sasaran mutu serta dibuktikan dengan kinerja lulusan di masyarakat sesuai dengan profesi dan bidang ilmu (BAN-PT, 2008).

1.5 Akreditasi

Akreditasi program studi merupakan proses evaluasi dan penilaian secara komprehensif atas komitmen program studi terhadap mutu dan kapasitas penyelenggaraan program tridarma perguruan tinggi, guna menentukan kelayakan program studi untuk menyelenggarakan program akademiknya (BAN-PT, 2008). Kriteria untuk mengevaluasi dan menilai komitmen tersebut dijabarkan dalam sejumlah standar akreditasi beserta parameterinya.

1.6 Statistik

Secara umum, statistik adalah suatu metode ilmiah dalam mengumpulkan, mengklasifikasikan, meringkas, menyajikan, menginterpretasikan, dan menganalisis data guna mendukung pengambilan kesimpulan yang valid dan berguna sehingga dapat menjadi dasar pengambilan keputusan yang masuk akal (Harinaldi, 2005:2).

Dalam pengertian terbatas, terminologi statistik digunakan untuk menyebutkan data itu sendiri, atau fakta berupa angka yang dihasilkan dari data, yang menggambarkan karakteristik suatu sampel. Dalam pengertian ini muncullah istilah-istilah seperti: statistik pegawai negeri, statistik kendaraan impor, statistik kecelakaan lalu lintas, dan sebagainya.

1.6.1 Populasi

Populasi adalah kumpulan dari keseluruhan pengukuran, objek, atau individu yang sedang dikaji (Harinaldi, 2005:2). Jadi pengertian populasi dalam

statistik tidak terbatas pada sekelompok/kumpulan orang-orang, namun mengacu pada seluruh ukuran, hitungan, atau kualitas yang menjadi fokus perhatian suatu kajian. Suatu pengamatan/survey terhadap seluruh anggota populasi disebut sensus.

1.6.2 Sampel

Sampel adalah sebagian atau subset (himpunan bagian) dari suatu populasi (Harinaldi, 2005:2). Populasi dapat berisi data yang besar sekali jumlahnya yang mengakibatkan tidak mungkin atau sulit dilakukan pengkajian terhadap seluruh data tersebut sehingga pengkajian dilakukan terhadap sampelnya saja.

1.6.3 Penarikan Sampel Berstrata dan Pengolahan Data

Sebelum mengolah data yang diperoleh dari kegiatan *tracer study*. Dilakukan penghitungan sampel minimum yang harus disiapkan untuk sebuah populasi. Populasi yang digunakan adalah populasi alumni yang lulus pada tahun-tahun tertentu. Populasi tersebut kemudian dibagi menjadi strata. Pembagian menjadi strata dilakukan berdasarkan jurusan alumni.

Penarikan sampel acak berstrata merupakan suatu metode di mana populasi yang berukuran N , dibagi-bagi menjadi sub-sub populasi yang masing-masing terdiri atas N_1, N_2, N_3, N_L elemen (Suryana, 2008). Setelah strata (sub populasi) terbentuk, penarikan sampel dari masing-masing strata dilakukan secara terpisah. Ukuran sampel yang ditarik masing-masing strata sebesar n_1, n_2, n_3, n_L . Pada penerapan rancangan sampel berstrata perlu diperhatikan variabel apa yang digunakan sebagai dasar pembentukan strata, alokasi sampel pada masing-masing

strata, dan ukuran sampel yang diperlukan untuk menduga statistik dengan presisi yang dikehendaki.

Metode ini dipilih karena dapat memperoleh nilai estimasi dengan presisi yang lebih tinggi untuk setiap strata maupun untuk populasi secara keseluruhan. Selain itu setiap strata dapat dianggap sebagai populasi tersendiri sehingga bisa saja menentukan presisi yang dikehendaki pada setiap strata dan disajikan tersendiri. Secara administratif pelaksanaannya relatif akan lebih mudah.

Untuk menentukan jumlah sampel minimal untuk seluruh populasi dapat menggunakan rumus:

$$n = \frac{N}{1 + Ne^2}$$

Di mana:

n : ukuran sampel keseluruhan

N : ukuran populasi

e : *bound of error* atau besarnya akurasi yang diinginkan dengan derajat keyakinan tertentu.

Sedangkan untuk menentukan jumlah sampel minimal untuk strata yang ke- i dapat menggunakan rumus:

$$n_i = \frac{N_i}{N} \times n$$

Di mana:

n_i : ukuran sampel pada strata ke- i

N_i : ukuran populasi pada strata ke- i

N : ukuran populasi

n : ukuran sampel keseluruhan

1.6.4 Variabel Nominal

Variabel nominal adalah skala yang hanya digunakan untuk memberikan kategori saja (Suliyanto, 2006:86). Kategori tersebut bersifat *mutually exclusive*. Artinya, jika satu indikator sudah masuk pada satu kategori maka tidak mungkin masuk ke kategori yang lain. Variabel nominal merupakan variabel yang memiliki tingkatan paling rendah di dalam sebuah riset.

1.6.5 Uji Chi-Square

Uji *chi-square* merupakan suatu ukuran mengenai perbedaan yang terdapat antara frekuensi yang diobservasi dan yang diharapkan (Spiegel, 1996:271). Uji *chi-square* berguna untuk menguji hubungan atau pengaruh dua buah variabel nominal dan mengukur kuatnya hubungan antara variabel yang satu dengan variabel nominal yang lainnya. *Chi-square* memiliki karakteristik antara lain:

1. Nilai *chi-square* selalu positif.
2. Terdapat keluarga distribusi *chi-square*, yaitu distribusi *chi-square* dengan $DK=1, 2, 3$, dan seterusnya.
3. Bentuk distribusi *chi-square* adalah menjulur positif.

Rumus *chi-square*:

$$\chi^2 = \left[\frac{\sum (f_o - f_e)^2}{f_e} \right]$$

Di mana:

χ^2 : Nilai *chi-square*.

f_e : Frekuensi yang diharapkan.

f_0 : Frekuensi yang didapatkan/diamati.

Uji *chi-square* dipilih karena menurut Hasan (2011), uji tersebut cocok digunakan untuk mengukur hubungan antara dua buah variabel. Selain itu uji *chi-square* dapat mengukur seberapa kuat hubungan variabel tersebut. Sehingga dapat diketahui variabel mana saja yang memiliki keterkaitan erat dengan kepuasan lulusan dan kompetensi lulusan di dunia kerja.

Sebagai contoh, uji *chi-square* digunakan untuk menguji apakah ada hubungan yang signifikan terhadap jurusan dengan penghasilan yang diperoleh pada saat pertama kali bekerja. Data yang diperoleh dari hasil kegiatan *tracer study* adalah sebagai berikut:

Tabel 2.1 Data Hasil *Tracer Study*

Alumni Sistem Informasi yang mendapatkan gaji pertama di bawah 1 juta adalah sebanyak 20 orang.
Alumni Sistem Komputer yang mendapatkan gaji pertama di bawah 1 juta adalah sebanyak 14 orang.
Alumni Sistem Informasi yang mendapatkan gaji pertama antara 1 dan 3 juta adalah sebanyak 116 orang.
Alumni Sistem Komputer yang mendapatkan gaji pertama antara 1 dan 3 juta adalah sebanyak 102 orang.
Alumni Sistem Informasi yang mendapatkan gaji pertama di atas 3 juta adalah sebanyak 14 orang.
Alumni Sistem Komputer yang mendapatkan gaji pertama di atas 3 juta adalah sebanyak 34 orang.

Setelah diketahui data-data tersebut, maka pengujian dengan *chi-square* dapat dilakukan dengan beberapa langkah berikut ini.

Uji Hipotesis:

H_0 : Tidak terdapat hubungan yang signifikan antara jurusan dengan penghasilan yang pertama.

H_1 : terdapat hubungan yang signifikan antara jurusan dengan penghasilan yang pertama.

$$\alpha = 0,05$$

Uji Statistik:

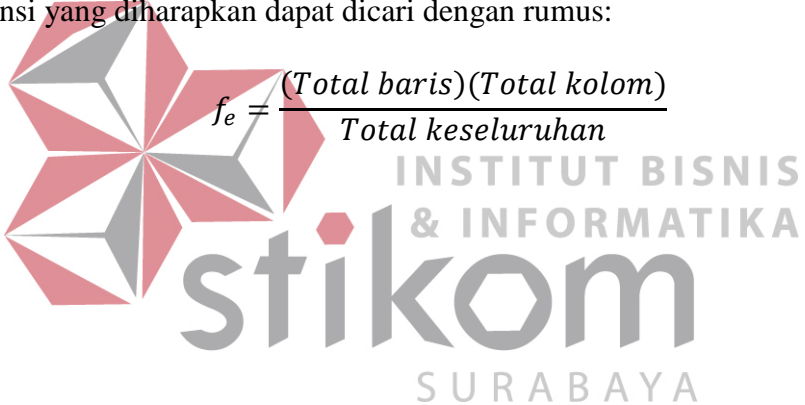
Setelah hipotesis dibuat, buat tabel kontingensi.

Tabel 2.2 Tabel Kontingensi

Jurusan	Gaji Pertama (juta)			Total
	Di bawah 1	Antara 1 dan 3	Di atas 3	
Sistem Informasi	20	116	14	150
Sistem Komputer	14	102	34	150
TOTAL	34	218	48	300

Setelah tabel kontingensi dibuat, cari frekuensi yang diharapkan (f_e).

Frekuensi yang diharapkan dapat dicari dengan rumus:

$$f_e = \frac{(Total\ baris)(Total\ kolom)}{Total\ keseluruhan}$$


Logo of Institut Bisnis & Informatika (STIKOM) SURABAYA. The logo features a stylized red and white geometric design on the left, with the text 'stikom' in a large, bold, lowercase font, and 'INSTITUT BISNIS & INFORMATIKA' and 'SURABAYA' in smaller, uppercase fonts above and below it, respectively.

Tabel 2.3 Tabel Kontingensi dan Hasil Perhitungan f_e

Jurusan	Gaji Pertama (Juta)						TOTAL	
	Di bawah 1		Antara 1 dan 3		Di atas 3		f_0	f_e
	f_0	f_e	f_0	f_e	f_0	f_e		
SI	20	17	116	109	14	24	150	150
SK	14	17	102	109	34	24	150	150
TOTAL	34	34	218	218	48	48	300	300

Langkah selanjutnya adalah dengan menghitung nilai *chi-square* (χ^2).

$$\chi_{hitung}^2 = \left[\frac{\sum (f_0 - f_e)^2}{f_e} \right]$$

$$\chi_{hitung}^2 = \frac{(20 - 17)^2}{17} + \frac{(116 - 109)^2}{109} + \frac{(14 - 24)^2}{24} + \frac{(14 - 17)^2}{17}$$

$$+ \frac{(102 - 109)^2}{109} + \frac{(34 - 24)^2}{24}$$

$$\chi_{hitung}^2 = 0,529 + 0,45 + 4,167 + 0,529 + 0,45 + 4,167$$

$$\chi_{hitung}^2 = 10,292$$

Setelah itu tentukan nilai χ^2 pada tabel distribusi χ^2 . Diasumsikan taraf signifikansi (α) adalah 0,05.

$$Df = (\text{Baris}-1)(\text{Kolom}-1)$$

$$= (2-1)(3-1)$$

$$= 2$$

Maka didapatkan $\chi^2_{tabel} = 5,991$.

Daerah kritis:

H_0 ditolak jika $\chi_{hitung}^2 > \chi_{tabel}^2$.

Kesimpulan:

H_0 ditolak artinya terdapat hubungan yang signifikan antara jurusan dengan penghasilan pertama yang didapatkan.

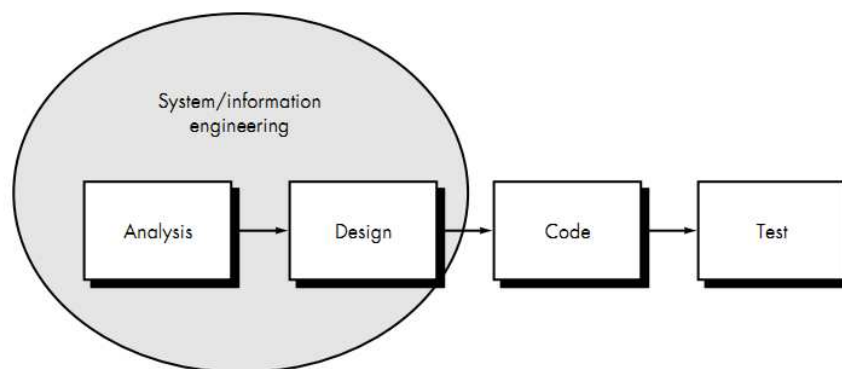
1.7 Website

Website merupakan sebuah sistem informasi yang terdistribusi berbasis internet (Wang, 2004:5). Siapapun yang terhubung dengan internet dapat dengan mudah memperoleh informasi dengan memberikan sebuah alamat *web* atau dengan menekan sebuah tombol *mouse*. *Website* merupakan alat yang handal untuk menyebarkan informasi dan membuatnya tersedia selama 24 jam dalam sehari dan 7 (tujuh) hari dalam seminggu. Informasi juga dapat diperoleh dari pengguna *website* dan pelanggan melalui formulir *online*. Administrator dapat mengendalikan dan memperbarui isi *website* dari manapun. Dengan semua keistimewaan ini membuat *website* menjadi alat yang handal sebagai komunikasi massal, e-bisnis, dan *e-commerce*.

Tidak ada kontrol atau administrasi terpusat pada sebuah *website*. Siapapun memungkinkan untuk menaruh materi ke dalam *website* dan mendapatkan informasi darinya. Sebuah *website* terdiri dari banyak sekali jenis dokumen yang terdapat pada komputer di seluruh dunia. Dokumen-dokumen tersebut disiapkan dalam format khusus dan didapatkan melalui program *server* pada setiap komputer yang menyediakan layanan *web*. setiap dokumen web dapat mengandung tautan ke dokumen yang lain yang disediakan oleh *server* yang berbeda.

1.8 Waterfall Model

Waterfall model merupakan salah satu model yang digunakan untuk membangun sebuah sistem. *Waterfall model* menawarkan pendekatan yang berurutan dan sistematis dalam pengembangan perangkat lunak. Seperti yang terlihat pada gambar 2.1, setiap pekerjaan saling bergantung dengan pekerjaan sebelumnya sehingga sebuah *task* tidak dapat dilakukan apabila *task* yang lain belum diselesaikan.



Gambar 2.1 *The linear sequential model/waterfall model* (Pressman, 2001:29)

Terdapat beberapa tahapan yang harus dilakukan di dalam *Waterfall Model* (Pressman, 2001), yaitu:

1. *Software Requirement Analysis*

Merupakan layanan, batasan dan tujuan dari sistem yang dibuat dengan mengkonsultasikannya bersama para pengguna sistem. Hal tersebut didefinisikan secara detail dan ditampilkan sebagai spesifikasi dari sistem.

2. *Design*

Proses desain sistem membagi kebutuhan sistem akan perangkat lunak dan perangkat keras. Hal tersebut membangun sistem keseluruhan. Desain

perangkat lunak meliputi identifikasi dan penjabaran abstraksi sistem perangkat lunak dasar dan keterhubungannya.

3. *Code Generation*

Selama tahapan ini, desain perangkat lunak direalisasikan sebagai sekumpulan program atau unit program. Unit testing meliputi verifikasi bahwa setiap unit telah memenuhi spesifikasinya.

4. *Testing*

Unit-unit program individual digabungkan (*integrated*) dan diuji coba (*tested*) sebagai sebuah sistem lengkap untuk memastikan bahwa kebutuhan-kebutuhan perangkat lunak telah terpenuhi. Setelah pengujian, sistem perangkat lunak disampaikan kepada pelanggan.

5. *Support*

Sistem diinstal dan digunakan secara praktikal. Pemeliharaan meliputi perbaikan kesalahan yang tidak diketahui pada tahapan sebelumnya, memperbaiki implementasi unit sistem dan meningkatkan layanan sistem ketika terdapat kebutuhan baru.

1.9 Unified Modeling Language (UML)

UML adalah sebuah bahasa pemodelan visual yang memungkinkan pengembang sistem untuk membuat cetak biru (*blueprints*) yang menggambarkan suatu visi dalam sebuah standar, dengan cara yang mudah dimengerti, dan menyediakan mekanisme untuk membagikan dan mengkomunikasikan visi tersebut kepada orang lain secara efektif (Schmuller, 2004:7). Notasi UML dibuat sebagai kolaborasi dari Grady Booch, DR. James Rumbaugh, Ivar Jacobson,

Rebecca Wirfs-Brock, Peter Yourdon, dan lainnya. Penggabungan beberapa metode menjadi UML dimulai tahun 1993.

1.9.1 Diagram Use Case

Diagram *use case* menyajikan interaksi antara *use case* dan aktor dalam sistem yang akan dikembangkan. *Use case* sendiri adalah fungsionalitas atau persyaratan-persyaratan sistem yang harus dipenuhi oleh sistem yang akan dikembangkan tersebut menurut pandangan pemakai sistem. Sedangkan aktor bisa berupa orang, peralatan atau sistem lain yang berinteraksi terhadap sistem yang akan dibangun (Sholiq, 2010).

1.9.2 Flow Of Events

Flow of events digunakan untuk mendokumentasikan aliran logika dalam *use case*, yang menjelaskan secara rinci apa yang pemakai akan lakukan dan apa yang sistem itu sendiri lakukan (Sholiq, 2010:102). Namun pada bagian ini tidak dijelaskan apakah sistem dibangun menggunakan bahasa pemrograman Java, Visual Basic, atau PHP.

1.9.3 Sequence Diagram

Diagram sekuensial (*sequence diagram*) adalah diagram yang digunakan untuk menunjukkan alur (*flows*) fungsionalitas yang melalui sebuah *use case* yang disusun dalam urutan waktu (Sholiq, 2010:23). Setiap diagram sekuensial merepresentasikan satu *flow* dari beberapa *flow* di dalam *use case*.

1.9.4 *Class Diagram*

Diagram kelas menunjukkan interaksi antar kelas-kelas di dalam sistem. Kelas juga dapat dianggap sebagai cetak biru dari objek-objek di dalam sistem. Sebuah kelas dibuat dalam bentuk bujur sangkar yang terbagi dalam tiga bagian. Bagian pertama menunjukkan nama kelas. Bagian kedua menunjukkan anggota kelas yang memuat informasi atau atribut. Bagian ketiga menunjukkan operasi-operasi dari sebuah kelas. Operasi dari sebuah kelas adalah tingkah laku yang disediakan oleh kelas (Sholiq, 2010:28).

1.9.5 *Diagram Komponen*

Diagram komponen adalah diagram yang menunjukkan model secara fisik atau komponen perangkat lunak pada sistem dan hubungannya antar mereka (Sholiq, 2010:30). Komponen terhubung oleh garis putus-putus yang menampilkan hubungan dependensi antar komponen. Relasi yang terjadi antar komponen hanya satu tipe relasi yaitu dependensi yang menunjukkan ketergantungan *compile-time* dan *run-time* antara komponen-komponen tersebut.

1.9.6 *Deployment Diagram*

Diagram deployment menampilkan rancangan fisik jaringan tempat berbagai komponen akan diletakkan. Peralatan fisik tersebut disebut node yang dihubungkan satu sama lain dengan menggunakan garis yang disebut *path* (Sholiq, 2010:31). Diagram deployment digunakan oleh manajer proyek, arsitek sistem, atau bagian distribusi untuk memahami rancangan fisik sistem, apa saja dan di mana saja terdapat subsistem yang akan dibuat.

1.10 *Black Box Testing*

Black box testing merupakan pengujian yang dilakukan tanpa pengetahuan detail struktur internal dari sistem atau komponen yang diuji (Romeo, 2003). Pengujian ini disebut juga sebagai *behavioral testing*, *specification-based testing*, *input/output testing* atau *functional testing*. *Black box testing* dilakukan pada tahap akhir dan berfokus pada domain informasi.

1.11 *Konsep Model-View-Controller (MVC)*

Model-View-Controller (MVC) merupakan sebuah pola desain suatu *software development*. MVC adalah sebuah pendekatan untuk memisahkan aplikasi yang dibuat ke dalam 3 (tiga) bagian: *Models*, *Views*, dan *Controller* (Griffiths, 2010). MVC di sini bertujuan agar kode-kode yang telah dibuat dapat digunakan kembali.

Model merepresentasikan data apapun yang akan digunakan oleh aplikasi. *View* adalah informasi yang sedang ditampilkan di dalam layar kepada user melalui browser yang digunakan. *View* dapat berupa berkas HTML yang mengandung kode-kode PHP untuk membangun *template website* tersebut. Sedangkan *Controller* merupakan *business logic* dari aplikasi yang dibuat. *Controller* bekerja sebagai perantara antara *Models* dan *Views*. *Controller* akan merespon permintaan HTTP dan membentuk halaman *web* yang diminta.

1.12 *CodeIgniter*

CodeIgniter merupakan sebuah *framework* aplikasi *web* yang bersifat sumber terbuka untuk bahasa PHP (Griffiths, 2010). CodeIgniter memiliki dokumentasi yang relatif lengkap dibandingkan dengan *framework* yang lain.

CodeIgniter kompatibel dengan PHP4 maupun PHP5 sehingga membuatnya dapat dijalankan pada banyak *web hosting* pada umumnya.

CodeIgniter adalah sebuah *framework* yang menerapkan konsep *Model-View-Controller* (MVC). Dengan demikian CodeIgniter akan mengorganisir aplikasi yang akan dibuat ke dalam 3 (tiga) bagian: *Models* – *layer* abstraksi database, *Views* – berkas *template* untuk tampilan halaman *web*, dan *Controller* – *business logic* dari aplikasi yang akan dibuat. Konsep tersebut sangat berguna untuk aplikasi yang terkoneksi dengan *database* karena hanya akan membutuhkan satu koneksi ketika diperlukan.

CodeIgniter juga telah mengimplementasikan pola *Active Record* yang menyederhanakan penulisan sintaks SQL yang rumit menjadi lebih mudah untuk dibaca. *Active Record* juga memudahkan pengembang dalam hal perubahan *driver database*. Dengan adanya fasilitas tersebut, pengembang yang ingin mengganti *databasenya* dari MySQL ke Oracle tidak perlu mengganti lagi kode-kode program yang telah dibuat sebelumnya.

1.13 Oracle Database

Oracle Database merupakan sebuah *object-relational database*. Oracle Database didesain untuk menyimpan dan mengelola informasi yang berskala *enterprise*. Oracle Database dapat menghemat biaya manajemen dan menyediakan layanan dengan kualitas yang tinggi (Greenberg, 2004).

1.14 Netbeans

Netbeans adalah sebuah proyek sumber terbuka yang didedikasikan untuk menyediakan produk *software development* handal yang ditujukan pada

kebutuhan dari pengembang, pengguna, dan bisnis (Netbeans, 2012). Dua basis produk proyek Netbeans adalah Netbeans IDE dan Netbeans Platform. Keduanya gratis untuk siapapun yang akan membangun sistem yang akan dikomersilkan sekalipun.

