BAB III

OBJEK DAN METODE PENELITIAN

3.1 Objek Penelitian

Objek penelitian ini berisikan tentang gambaran tempat penulis melakukan penelitian, diantaranya tentang sejarah, struktur organisasi dan deskripsi pekerjaan pada Program Studi Sistem Informasi UNIKOM.

3.1.1 Sejarah Singkat Perusahaan

Universitas Komputer Indonesia (UNIKOM) secara resmi berdiri pada hari Selasa, tanggal 8 Agustus 2000 berdasarkan Surat Keputusan Menteri Pendidikan Nasional nomor 126/D/0/2000.

Awalnya dimulai pada bulan Juli tahun 1994 ketika didirikan Lembaga Pendidikan Komputer Indonesia Jerman, disingkat LPKIG, bertempat di jalan Dipati Ukur 102 Bandung . Dengan 1 ruang kelas berkapasitas 50 orang dan 1 laboratorium komputer dengan 25 unit komputer, Lembaga ini membuka program pendidikan 1 tahun dengan 5 program studi yaitu Ahli Komputer Aplikasi Bisnis, Ahli Komputer Keuangan & Perbankan, Ahli Komputer Akuntansi & Perpajakan, Ahli Komputer Manajemen & Pemasaran dan Sekretaris Eksekutif. Jumlah peserta pendidikan pada tahun pertama ini sebanyak 233 siswa.

Pada tahun kedua, 1995, dibuka jenjang pendidikan 3 tahun untuk memenuhi animo siswa tahun pertama yang ingin memperdalam ilmunya, disamping pemikiran jangka panjang pengembangan institusi. Pada tahun ini juga

dibuka program studi baru, meliputi : Ahli Komputer Teknik Informatika, Ahli Komputer Manajemen Informatika dan Sekretaris Eksekutif. Ruang kelas ditambah menjadi 2 buah dan laboratorium komputer menjadi 2 buah dengan jumlah siswa sebanyak 457 orang.

Pada tahun ketiga, 1996, dilakukan penambahan gedung kuliah baru bertempat di jalan Dipati Ukur 116 (gedung FISIP sekarang), sekaligus pemindahan pusat administrasi dan perkantoran. Digedung baru ini dilakukan penambahan 1(satu) Lab. Komputer, 5(lima) Ruang Kuliah, Ruang Dosen dan Ruang Kemahasiswaan. Jumlah siswa dari tahun 1996 hingga tahun 1998 bertambah dari 632 orang menjadi 1184 orang.

Pada tahun kelima, 1998, dimulai pembangunan Kampus baru (Gedung Rektorat /Kampus-1 sekarang) berlantai 6(enam) di jalan Dipati Ukur 114. Pembangunan Kampus baru ini dapat diselesaikan pada bulan Agustus 1999, sehingga pada awal perkuliahan bulan September 1999 telah dapat digunakan.

Mencermati dinamika peserta didik dan pengembangan Institusi kedepan, pada tanggal 24 Desember 1998 dibentuklah Yayasan Science dan Teknologi dan dilanjutkan dengan pengajuan pendirian STIMIK IGI dan STIE IGI ke DIKTI.

Pada bulan Juli 1999 STIE IGI diresmikan dengan keluarnya SK Mendiknas no. 119/D/O/1999 dengan 5 program studi : Akuntansi S1, Manajemen S1, Manajemen Pemasaran D3, Keuangan Perbankan D3 serta Akuntansi D3.

Pada bulan Agustus 1999 STIMIK IGI diresmikan dengan keluarnya SK Mendiknas no. 143/D/O/1999 dengan 5 program studi : Teknik Informatika S1, Manajemen Informatika D3, Teknik Komputer D3, Komputerisasi Akuntansi D3 serta Teknik Informatika D3.

Agar Sistem Pendidikan lebih Efisien, Efektif, Produktif dengan Struktur Organisasi yang lebih baik, enam bulan kemudian dilakukan usulan ke DIKTI untuk melakukan Merger kedua Sekolah Tinggi diatas menjadi Universitas.

Pada hari Selasa, tgl. 8 Agustus 2000 keluarlah SK MENDIKNAS no. 126/D/O/2000 atas Universitas Komputer Indonesia yang disingkat dengan nama UNIKOM.Pada SK tersebut sekaligus diijinkan dibukanya 11 program studi baru : Teknik Komputer S1, Manajemen Informatika S1, Teknik Industri S1, Teknik Arsitektur S1, Perencanaan Wilayah dan Kota S1, Ilmu Hukum S1, Ilmu Komunikasi S1, Ilmu Pemerintahan S1, Desain Interior D3, Desain Komunikasi Visual S1 dan Desain Komunikasi Visual D3.

Sejak berdirinya pada tahun 2000, setiap tahunnya UNIKOM menerima ± 2.000 mahasiswa baru. Terakhir pada tahun 2009 yang lalu diterima sebanyak 3.108 mahasiswa baru. Hingga tahun akademik 2009/2010 terdapat 6 Fakultas dan 23 Program Studi di UNIKOM dengan jumlah mahasiswa sebanyak 15.000 orang yang berasal dari berbagai pelosok tanah air dan dari luar negeri yang sedang menempuh pendidikan di UNIKOM.

3.1.2 Visi dan Misi

1. Visi Perusahaan

   Visi dari Program Studi Sistem Informasi UNIKOM adalah menjadikan Program Studi yang berkompeten, unggul, terdepan dalam bidang Teknologi dan sistem informasi serta menghasilkan lulusan yang berprestasi, berjiwa entrepreneur dan mampu berkompetisi di era global.

2. Misi Perusahaan

   Misi dari Program Studi Sistem Informasi UNIKOM adalah:

   a. Menyelenggarakan pendidikan akademik dan pengajaran profesional dengan memberikan pengetahuan teoritis maupun praktis yang efektif dan efisien dengan pemanfaatan teknologi informasi agar menghasilkan lulusan yang kompeten di bidang Sistem Informasi dan mampu mengaplikasikannya sesuai dengan kebutuhan pengguna (*users*), serta mampu menciptakan lapangan kerja sendiri secara tangguh dan mandiri.

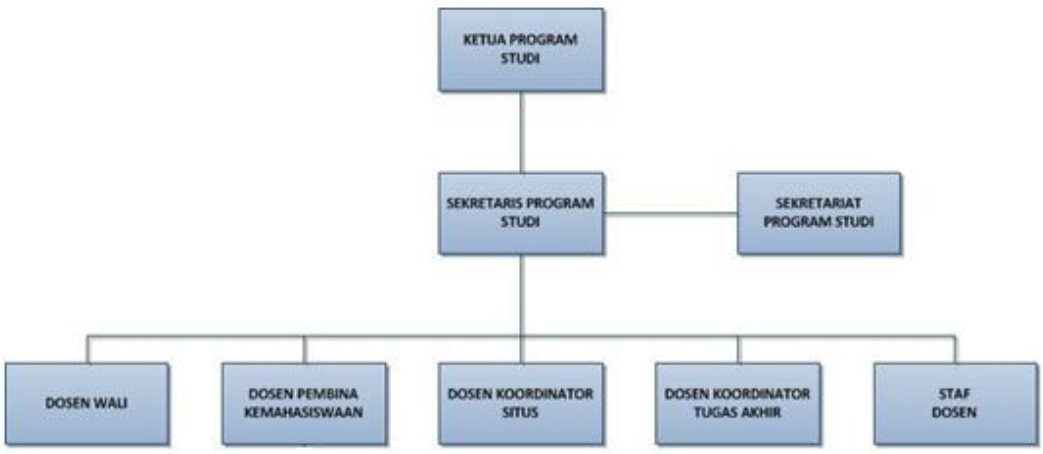
   b. Meningkatkan kualitas keterampilan mahasiswa untuk jaminan standar kualitas lulusan yang memiliki dedikasi, bermoral, berintegrasi, berwawasan dan mampu beradaptasi dengan perubahan lingkungan baik nasional maupun global.

   c. Melaksanakan program pengabdian kepada masyarakat di bidang sistem Informasi sebagai tanggung jawab sosial dengan melibatkan partisipasi aktif sivitas akademika.

d. Melaksanakan kolaborasi dengan berbagai pihak dalam bidang pendidikan, penelitian dan pengabdian masyarakat di tingkat nasional dan internasional secara efektif, efisien dan berkesinambungan untuk pengembangan dan keberlangsungan pendidikan.
e. Menjadikan tempat Uji Kompetensi dalam bidang Sistem Informasi dan teknologi Informasi.

3.1.3 Struktur Organisasi

Struktur organisasi mempunyai arti penting karena struktur organisasi merupakan bentuk atau pola formal kegiatan dan hubungan antara berbagai bagian-bagian didalam suatu peusahaan. Dengan mengetahui struktur organisasi dapat diperoleh gambaran tentang bagian-bagian yang ada di dalamnya, apa peranan masing-masing bagian tersebut dan wewenang serta tanggung jawabnya dalam melaksanakan tugasnya.



Gambar 3. 1 Struktur Organisasi Program Studi Sistem Informasi Unikom

(Sumber http://if.unikom.ac.id/profile.php)

3.1.4 Deskripsi Tugas

Deskripsi tugas atau job description adalah suatu rincian yang menunjukan posisi, tanggung jawab, wewenang, fungsi dan tugas tugas yang harus dilakukan. Deskripsi tugas perlu dibuat agar masing – masing bagian mengerti akan kedudukan didalam organisasi. Adapun uraian jabatan pada Program Studi Sistem Informasi UNIKOM, yaitu :

a. Ketua Program Studi

   Bertanggung jawab pada program yang berjalan untuk program studi yang dikepalai dan mengambil keputusan-keputusan pada program studi yang dikepalai.

b. Sekretaris Program Studi

   Membantu ketua program studi dalam menjalankan tugasnya

c. Sekretariat Program Studi

   Mengurus segala keperluan administrasi perkuliahan pada program studinya

d. Dosen Wali

   Bertanggung jawab untuk setiap mahasiswa walinya dan membantu mahasiswa walinya yang dalam mengurus administrasi perkuliahan

e. Dosen Pembina Kemahasiswaan

   Bertanggung jawab untuk membina organisasi-organisasi kemahasiswaan yang ada pada program studi

f. Dosen Koordinator Situs

   Bertanggung jawab untuk maintenance situs program studi

g. Dosen Koordinator Tugas Akhir

   Bertanggung jawab untuk mengkoordinasi kegiatan Tugas Akhir pada program studi

h. Staff Dosen

   Bertanggung jawab untuk mengajarkan mata kuliah sesuai dengan jadwal yang ditentukan

3.2 Metode Penelitian

Metode penelitian yang digunakan adalah metode deskriptif dengan jenis penelitian studi kasus. Metode deskriptif adalah suatu metode dalam meneliti status kelompok manusia, suatu objek, suatu set kondisi, suatu sistem pemikiran ataupun suatu peristiwa pada masa sekarang.

3.2.1 Desain Penelitian

Desain penelitian merupakan perpaduan antara keputusan dan revisi dimana suatu keputusan yang diambil selalu diiringi dengan pengaruh adanya keseimbangan dalam proses.

3.2.2 Jenis Dan Metode Pengumpulan Data

Pengumpulan data terdapat dua sumber data yaitu sumber data primer dan sumber data sekunder.

3.2.2.1 Sumber Data Primer (Wawancara, Observasi, Kuisioner)

Adapun langkah-langkah yang dilakukan oleh penulis dalam pengumpulan data diantaranya adalah :

1. Observasi

   Observasi yaitu pengamatan dengan cara pemusatan terhadap proses pembuatan jadwal kuliah, baik dalam penentuan prioritas-prioritas dalam pembuatan jadwal kuliah serta syarat-syarat tertentu yang harus dipenuhi pada jadwal kuliah yang ada pada UNIKOM, terkhusus Prodi Sistem Informasi.

2. Wawancara

   Wawancara dengan ketua Prodi dan juga sekretariat jurusan untuk pengambilan data perihal dalam pembuatan jadwal kuliah pada Prodi Sistem Informasi di UNIKOM.

3.2.2.2 Sumber Data Sekunder (dokumentasi)

Dalam pengambilan data sekunder ini penulis mengambil data-data yang berhubungan dengan proses penjadwalan kuliah di Prodi Sistem Informasi di UNIKOM serta literature atau bahan lain untuk dijadikan bahan dalam penyusunan skripsi.

Dokumentasi yang dibutuhkan antara lain:

1. Jadwal kuliah pada semester-semester sebelumnya
2. Data mengenai syarat dan prioritas yang harus ada pada penyusunan jadwal kuliah di Prodi Sistem Informasi UNIKOM.
3. Profil perusahaan serta struktur organisasi pada Prodi Sistem Informasi

4. Literatur dan artikel yang dibutuhkan terkait dalam pengembangan aplikasi penjadwalan kuliah pada Program Studi Sistem Informasi UNIKOM.

3.2.3 Metode Pendekatan dan Pengembangan Sistem

Metode pendekatan dan pengembangan sitem merupakan sistematis dari proses penelitian yang dilakukan guna menyelesaikan permasalahan yang ada pada awal perencanaan penelitian hingga tercapainya penelitian tersebut dan pengembangan sistem yang diinginkan.

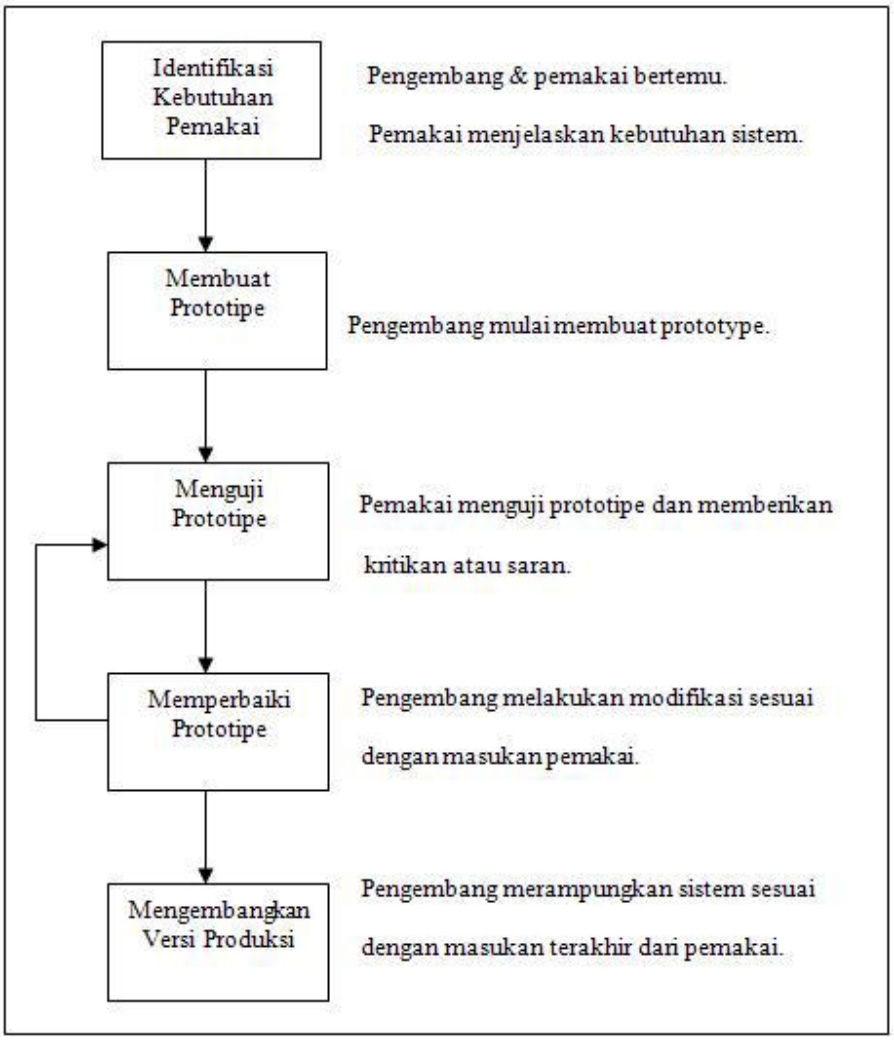
3.2.3.1 Metode Pendekatan Sistem

Aplikasi yang akan dibangun merupakan aplikasi intern, digunakan untuk membuat jadwal kuliah pada Prodi Sistem Informasi di UNIKOM, di mana pengguna yang akan memakai applikasi tersebut adalah satu jenis pengguna (ditujukan untuk staff Tata Usaha atau Sekretariat Jurusan).

Desain penelitian ini dimodelkan dengan menggunakan pemodelan *prototype,* yakni metode dalam pengembangan perangkat lunak dengan cepat dan bertahap agar dapat segera dievaluasi oleh pengguna. Penulis menggunakan metode pengembangan perangkat lunak *prototype* dengan harapan agar penulis akan dapat lebih mudah dalam merancang dan memenuhi kebutuhan dari pengguna dan dapat diterima dan digunakan pada prakteknya. Penulis menginginkan perancangan perangkat lunak yang telah dihasilkan dapat dipresentasikan kepada pengguna dan pengguna diberikan kesempatan untuk memberikan masukan-masukan, sehingga perangkat lunak yang dihasilkan

nantinya dapat dibangun sesuai dengan kebutuhan serta dapat digunakan pada prakteknya.

Metode *prototype* merupakan metode pengembangan perangkat lunak yang dirancang agar dapat menerima perubahan-perubahan dalam rangka menyempurnakan *prototype* yang sudah ada hingga akhirnya perangkat lunak tersebut dapat memenuhi kebutuhan pengguna dan siap untuk digunakan. Berikut merupakan gambaran metode pendekatan yang digunakan oleh penulis:



Gambar 3. 2 Pendekatan Prototype (Sumber, Abdul Kadir 2004)

Setiap metode pengembangan perangkat lunak pastilah memiliki kelebihan serta kekurangan masing-masing. Kelebihan dan kekurangan metode *prototype* antara lain:

1. Kelebihan
    a. Pendefinisian kebutuhan dari pengguna akan lebih jelas karena pengguna terlibat secara langsung dalam proses pengembangan.
    b. Meminimalisasi kesalahan yang mungkin terjadi, karena pada setiap versi *prototype* yang diberikan, pengguna dapat langsung mendeteksi kesalahan yang ada.
    c. Pengguna dapat meminta perubahan-perubahan untuk penyempurnaan aplikasi.
    d. Mempersingkat waktu yang dibutuhkan dalam pembangunan aplikasi secara keseluruhan.
    e. Penghematan biaya jika dibandingkan dengan metodologi pengembangan secara tradisional.
2. Kekurangan
    a. Perangkat lunak yang dibangun baru akan memberikan hasil yang optimal apabila pengguna sungguh-sungguh meluangkan waktu dalam pengembangan perangkat lunak.
    b. Pengembang yang terlalu terkonsentrasi pada pengembangan versi prototype sering mengakibatkan terbengkalainya pendokumentasian.
    c. Kekurangan waktu pengembangan dapat menyebabkan sistem yang tidak lengkap serta pengujian yang masih kurang.

d. Proses pengulangan pengembangan (*version*) dapat memberikan respon kurang baik dari pengguna serta mengakibatkan pengembangan memakan waktu yang lama.
e. Apabila pengguna selalu memiliki perubahan permintaan atau penyempurnaan, ada kemungkinan prototype tidak akan pernah mendapat finalisasi.

Berikut merupakan tahapan-tahapan yang dilakukan penulis dalam rangka mengembangkan perangkat lunak yang menggunakan mekanisme pengembangan aplikasi dengan metode *prototype:*

a. Penulis mengidentifikasi kebutuhan dari pengguna, agar penulis dapat melakukan perancangan mengenai perangkat lunak yang akan dibangun agar sesuai dengan yang harapan dari pengguna. Sebelum memulai perancangan, penulis menganalisis kebutuhan dari pengguna dengan cara melakukan pengumpulan data dengan observasi dan wawancara, serta membaca beberapa literatur mengenai algoritma yang akan digunakan.
b. Pada tahap kedua, penulis membuat prototype pertama untuk diperlihatkan pada pengguna model agar dapat dilakukan pengembangan selanjutnya.
c. Pada tahap ketiga, penulis melakukan uji coba perangkat lunak untuk dapat memastikan bahwa perangkat lunak dapat dengan baik dan benar serta sesuai dengan kebutuhan dari pengguna.
d. Pada tahap keempat, penulis dan pengguna akan menentukan apakah *prototype* perangkat lunak sudah dapat diterima dan sesuai kebutuhan, atau dilakukan perbaikan dan pengembangan berikut. Apabila harus dilakukan

pengembangan selanjutnya atau perbaikan, maka tahapan akan kembali ke tahap ke dua, yakni membuat prototype berikutnya untuk kemudian diuji lagi.

e. Pada tahap kelima (apabila perangkat lunak sudah sesuai dengan kebutuhan pengguna), penulis akan memberikan versi akhir pada pengguna agar dapat digunakan.

3.2.3.2 Alat Bantu Analisis dan Perancangan

Sesuai dengan pendekatan pengembangan yang diambil penulis, pendekatan berbasis objek, maka penulis akan menggambarkan karakteristik dari perangkat lunak yang akan dibangun dengan menggunakan pemodelan *Unified Modelling Language* (UML). UML merupakan pemodelan perangkat lunak yang digunakan untuk perangkat lunak yang dibangun menggunakan pendekatan berbasis objek. Keunggulan dari UML antara lain:

1. ***Uniformity.*** Dengan metodologi UML (atau berorientasi objek pada umumnya), para pengembang cukup menggunakan 1 metodologi dari tahap analisis hingga perancangan. Hal ini tidak bisa dilakukan dalam metodologi pengembangan terstruktur.Dengan perkembangan masa kini ke arah aplikasi GUI (*Graphical User Interface*), UML juga memungkinkan kita merancang komponen antar muka pengguna (*User Interface*) secara terintegrasi bersama dengan perancangan perangkat lunak sekaligus dengan perancangan basis data.

2. ***Understandibility.*** Dengan metodologi ini kode yang dihasilkan dapat diorganisasi kedalam kelas-kelas yang menghubungkan masalah sesungguhnya sehingga lebih mudah dipahami siapa pun juga.
3. ***Stability.*** Kode program yang dihasilkan relatif stabil sepanjang waktu sebab sangat mendekati permasalahan sesungguhnya dilapangan.
4. ***Reusability.*** Dengan metodologi berorientasi objek, dimungkinkan penggunaan ulang kode, sehingga pada gilirannyaakan sangat mempercepat waktu pengembangan perangkat lunak.

Sebagaimana yang telah dijelaskan diatas tentang UML, bahwa untuk mendapatkan banyak pandangan terhadap sistem informasi yang akan dibangun, UML menyediakan beberapa alat bantu diagram visual yang menunjukan berbagai aspek dalam sistem. Ada beberapa alat bantu diagram yang disediakan dalam UML antara lain :

1. ***Use Case Diagram***

   Use case mendepkripsikan interaksi tipikal antara para pengguna sistem dengan sistem itu sendiri, dengan member sebuah narasi tentang bagaimana sistem tersebut digunakan. *Use Case Diagram* menampilkan *actor* mana yang menggunakan *Use Case* mana, *Use Case* mana yang memasukan *Use Case* lain dan hubungan antara *actor* dan *Use Case*.

2. ***Activity Diagram***

   Diagram ini menjelaskan alur kerja suatu sistem. *Activity diagram* mirip dengan *state diagram* karena sejumlah aktifitas menggambarkan keadaan suatu proses dengan memperlihatkan urutan aktifitas yang dijalankan baik

berupa pilihan maupun paralel. Diagram ini juga berguna untuk menganalisis sebuah *use case* dengan menggambarkan aksi-aksi yang diperlukan dan kapan aksi-aksi tersebut dijalankan. Selain itu, *activity diagram* dapat menjelaskan urutan algoritma yang kompleks dan memodelkan sejumlah aplikasi dengan proses paralel.

3. ***Collaboration Diagram***

   Secara fungsional digram ini hampir mirip dengan *sequence diagram*. *Collaboration diagram* memfokuskan pada interaksi dan hubungan diantara sekumpulan objek yang berkolaborasi. Hubungan-hubungan tersebut memperlihatkan objek aktual dan relasi yang terjadi diantara mereka yang digambarkan dengan sebuah garis. Di atas garis terdapat alur pesan yang dikirim objek yang berhubungan tersebut.

4. ***Class Diagram***

   Class diagram mendepkripsikan jenis-jenis objek dalam sistem dan berbagai macam hubungan statis yang terdapat diantara mereka. *Class diagram* juga menunjukan *property* dan operasi sebuah *class* dan batasan-batasan yang terdapat dalam hubungan-hubungan objek tersebut.

5. ***Sequence Diagram***

   Sequence diagram secara khusus menjabarkan *behavior* sebuah sekenario tunggal. *Sequence diagram* menunjukan sebuah objek contoh dan pasan-pesan yang melewati objek-objek dalam *use case*.

6. ***Component Diagram***

   Component diagram menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency*) di antaranya. Komponen piranti lunak adalah modul berisi *code*, baik berisi *source code* maupun *binary code*, baik *library* maupun *executable*, baik yang muncul pada *compile time*, *link time*, maupun *run time*. Umumnya komponen terbentuk dari beberapa *class* dan/atau *package*, tapi dapat juga dari komponen-komponen yang lebih kecil. Komponen dapat juga berupa *interface*, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain.

7. ***Deployment Diagram***

   Deployment/physical diagram menggambarkan detail bagaimana komponen di-*deploy* dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, server atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server, dan hal-hal lain yang bersifat fisikal. Sebuah *node* adalah server, *workstation*, atau piranti keras lain yang digunakan untuk men-*deploy* komponen dalam lingkungan sebenarnya. Hubungan antar *node* (misalnya TCP/IP) dan *requirement* dapat juga didefinisikan dalam diagram ini.

3.2.4 Pengujian *Software*

Metode pengujian perangkat lunak merupakan cara atau teknik yang dilakukan untuk memeriksa suatu perangkat lunak, apakah perangkat tersebut sudah sesuai dengan analisis, perancangan, serta kebutuhan dari pengguna.

Mekanisme yang dilakukan dalam proses pengujian ini umumnya merupakan tahapan akhir pada suatu pengembangan perangkat lunak guna menentukan apakah perangkat lunak tersebut sudah dapat dirilis atau belum, dan umumnya memiliki kemungkinan yang sangat tinggi dalam menemukan kesalahan pada perangkat lunak (terutama pada pengujian awal). Jenis pengujian perangkat lunak umumnya dibagi menjadi dua, yakni metode *white box testing* yang merupakan metode di mana perangkat lunak diuji berdasarkan alur algoritma yang ada pada penulisan program (membuka *source code* dari perangkat lunak) dan metode *black box testing* yakni metode diuji dengan mengikuti alur proses yang dijalankan oleh perangkat lunak secara fundamental tanpa memperhatikan logika dan penulisan algoritma pada internal perangkat lunak.

Pada karya ilmiah ini, penulis memilih untuk melakukan metode *black box testing. Black box testing* merupakan metode pengujian yang didasarkan pada proses dan spesifikasi dari perangkat lunak yang memiliki tujuan utama untuk menemukan :

1. Fungsi-fungsi yang salah atau hilang
2. Kesalahan interface
3. Kesalahan dalam struktur data atau akses database eksternal
4. Kesalahan performa
5. kesalahan inisialisasi dan terminasi