

BAB II

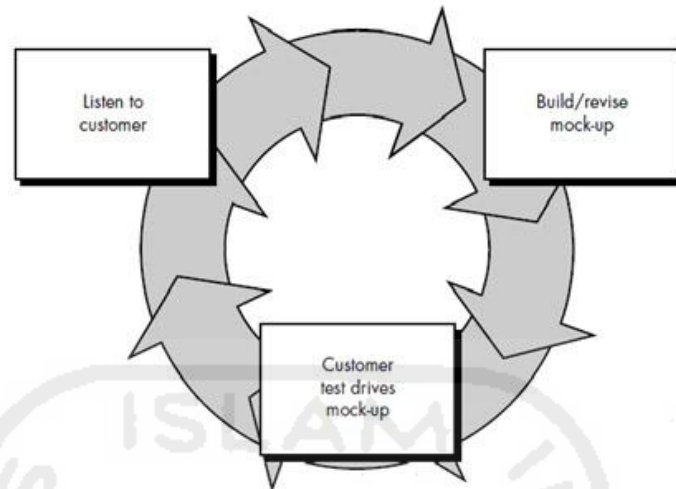
LANDASAN TEORI

2.1 *Prototyping*

Prototyping merupakan salah satu metode pengembangan perangkat lunak yang menggunakan pendekatan untuk membuat rancangan dengan cepat dan bertahap sehingga dapat segera dievaluasi oleh calon pengguna/klien. Dengan metode *prototyping* ini pengembang dan klien dapat saling berinteraksi selama proses pembuatan *prototype* sistem. Terkadang sering terjadi, klien hanya mendefinisikan secara umum apa yang dikehendaki tanpa menyebutkan proses masukan (*input*) dan keluaran (*output*) dari sistem yang akan dibuat. Untuk mengatasi ketidakselarasan tersebut maka harus dibutuhkan kerjasama yang baik di antara keduanya, sehingga pengembang akan mengetahui dengan benar apa yang dibutuhkan klien. Dengan demikian nantinya akan menghasilkan sebuah rancangan sistem yang interaktif sesuai dengan kebutuhan.

2.1.1 Metode *Prototype*

Metode *Prototype* menurut Pressman (2002:40), dimulai dengan mengumpulkan kebutuhan. Pengembang dan klien bertemu guna mendefinisikan obyektif keseluruhan dari perangkat lunak, mengidentifikasi segala kebutuhan dari segi *input* dan format *output* serta gambaran *interface*, kemudian dilakukan perancangan cepat. Dari hasil perancangan cepat tersebut nantinya akan dilakukan pengujian dan evaluasi. Penjelasan lengkap pada metode *prototype* akan dijelaskan melalui gambar pada halaman selanjutnya.



Gambar 2.1 Ilustrasi Metode *Prototype* (Pressman,2002).

Pada **Gambar 2.1** di atas terdapat tiga siklus yang akan dijelaskan sebagai berikut:

a. *Listen to Customer* (Mendengarkan Pelanggan)

Pada tahap ini merupakan identifikasi kebutuhan user, proses ini dilakukan agar penulis dapat memperoleh informasi mengenai permasalahan yang terjadi oleh klien. Data yang diperoleh dari permasalahan tersebut yang nantinya menjadi acuan untuk dilakukan proses pencarian solusi dan pengembangan pada tahap selanjutnya.

b. *Build and Revise Mock-up* (Membangun dan Memperbaiki *Prototype*)

Setelah kebutuhan sistem terkumpul, maka akan dilakukan proses perancangan *prototype* pada sistem yang diusulkan oleh *user*, yang mana tahap-tahapannya sebagai berikut:

- Perancangan proses-proses yang akan terjadi di dalam sistem, seperti, *input* (masukan), *output* (keluaran) dari sistem yang telah diusulkan.
- Perancangan UML (*Unified Modelling Language*), hal ini dilakukan untuk menspesifikasikan sistem tentang apa yang diperlukan dan bagaimana sistem tersebut direalisasikan.

Perancangan UML yang digunakan pada sistem ini meliputi: *Use-Case Diagram* dan *Activity Diagram*.

- Perancangan *Interface* (antarmuka) dan fitur yang dibutuhkan oleh klien (*User*).

c. *Customer Test Drives Mock-up* (Pengujian *Prototype*)

Pada tahap ini akan dilakukan pengujian terhadap *prototype* sistem yang telah dibuat, serta mengevaluasi apakah *prototype* sistem yang sudah dibuat sudah sesuai dengan yang diharapkan. Apabila dari hasil pengujian *prototype* tersebut belum memenuhi kebutuhan klien (*user*), maka pengembang akan melakukan proses perbaikan ulang *prototype* sampai *prototype* tersebut menjadi sistem yang final dan benar-benar diterima atau sesuai dengan keinginan *user*. Proses pengujian *prototype* sistem nantinya menggunakan teknik pengujian *black box*. Hasil dari pengujian *black box* tersebut nantinya akan dibahas pada pembahasan selanjutnya di Bab V mengenai hasil pengujian.

2.1.2 Keunggulan dan Kelemahan *Prototyping*

Achmad Rasul (2009) mengatakan bahwa suatu *prototyping* itu mempunyai beberapa keunggulan dan kelemahan.

Adapun beberapa keunggulan pada *prototyping*, antara lain sebagai berikut:

- a) Adanya komunikasi yang baik antara pengembang dan klien (*user*).
- b) Pengembang dapat bekerja lebih baik dalam menentukan kebutuhan klien.
- c) Klien berperan aktif dalam pengembangan sistem.
- d) Lebih menghemat waktu dalam pengembangan sistem.
- e) Penerapan menjadi lebih mudah karena klien mengetahui apa yang diharapkan.

Sedangkan kelemahan dari *prototyping* adalah sebagai berikut:

- a) Klien terkadang tidak melihat atau menyadari bahwa perangkat lunak yang ada belum mencantumkan kualitas perangkat lunak secara keseluruhan dan

juga belum memikirkan kemampuan pemeliharaan untuk jangka waktu lama.

- b) Pengembang biasanya ingin cepat menyelesaikan proyek. Sehingga menggunakan algoritma dan bahasa pemrograman yang sederhana untuk membuat *prototyping* lebih cepat tanpa memikirkan lebih lanjut bahwa program tersebut hanya merupakan cetak biru sistem.

2.2 Analisis Kebutuhan dan Perancangan Sistem

2.2.1 Analisis Kebutuhan

Analisis kebutuhan merupakan langkah untuk merencanakan dan mengidentifikasi kebutuhan sistem yang nantinya akan dibuat. Analisis juga dapat diartikan sebagai proses untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan, hambatan yang terjadi serta kebutuhan yang diharapkan sehingga dapat ditemukan solusi yang tepat. Tahap analisis merupakan tahap yang kritis dan penting, karena kesalahan di dalam tahap ini akan menyebabkan kesalahan di tahap selanjutnya (Jogiyanto, 2001).

Menurut Mahyuzir (1989) Analisis juga dijelaskan sebagai berikut:

1. Mengidentifikasi masalah-masalah kebutuhan pengguna.
2. Menyatakan secara spesifik sasaran yang harus dicapai untuk memenuhi kebutuhan pemakai.
3. Memilih alternatif metode pemecahan masalah yang paling tepat.

Setelah tahap analisis selesai dilakukan, maka nantinya akan terlihat jelas bagaimana gambaran tentang alur proses sistem. Proses selanjutnya yang harus dilakukan yaitu proses perancangan sistem. Proses tersebut akan dijelaskan pada penjelasan di bawah ini.

2.2.2 Perancangan Sistem

A. Definisi Perancangan Sistem

Menurut Burch dan Grudnitski (1986, p461) Perancangan sistem adalah penggambaran, perencanaan dan pembuatan sketsa atau pengaturan

dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi.

Perancangan sistem juga dapat diartikan sebagai berikut:

- a) Pendefinisian dari kebutuhan-kebutuhan fungsional.
- b) Persiapan untuk rancang bangun sebuah implementasi.
- c) Menggambarkan bagaimana suatu sistem terbentuk.

B. Tujuan Perancangan Sistem

Menurut Jogyanto (2001) tujuan utama dari tahap perancangan sistem yaitu untuk memenuhi kebutuhan para pemakai (*user*) dan memberikan gambaran yang jelas tentang rancang bangun secara lengkap kepada *programmer* serta pihak yang terlibat.

2.3 Mentoring Agama Islam

2.3.1 Pengertian Mentoring Agama Islam

Secara etimologi mentoring berasal dari kata dasar “mentor” yang merupakan istilah dari bahasa Yunani yang berarti pembimbing (Hidayana, 2010: 34). Pendapat lain menyatakan bahwa kata mentoring adalah kata yang berasal dari bahasa Arab yaitu *halaqah*, *liqa’* (lingkaran) atau *usrah*. Istilah mentoring (*liqa’*) biasanya digunakan untuk menggambarkan sekelompok kecil muslim yang secara rutin mengkaji ajaran Islam (Lubis, 2002: 1-2). Mentoring (*liqa’*) pada kelompok kecil berkisar antara 3-12 orang, sedangkan pada kelompok besar berkisar antara 12-48 orang. Satu orang bertindak sebagai narasumber, biasanya diistilahkan dengan mentor, murabbi atau Pembina (Lubis, 2002: 1).

Rusmiyati (2003) juga mendefinisikan Mentoring Agama Islam sebagai suatu kegiatan pembinaan pemuda pelajar yang berlangsung secara periodik dengan bimbingan seorang mentor.

Dalam suatu kelompok mentoring agama Islam terdapat istilah *mentor*. *Mentor* merupakan seorang yang ditunjuk oleh penanggungjawab kegiatan. Biasanya *mentor* merupakan kakak angkatan atau senior dari suatu tingkatan.

Selain itu terdapat istilah *Mentee* atau peserta mentoring, peserta mentoring dipimpin dan dibimbing oleh seorang *mentor* (Pembina).

Mentor bekerjasama dengan peserta mentoring untuk mencapai tujuan mentoring agama Islam yaitu terbentuknya muslim yang Islami. Dalam mencapai tujuan tersebut, *mentor* berusaha agar peserta hadir secara rutin dan aktif dalam pertemuan halaqah tanpa merasa jemu dan bosan dan juga menjaga kekompakan untuk mencapai tujuannya.

Dari pengertian mentoring tersebut di atas, maka dapat disimpulkan bahwa kegiatan mentoring agama Islam merupakan suatu kegiatan pembinaan pemuda pelajar (Mahasiswa) dengan jumlah peserta yang berkisar antara 3-12 orang yang berlangsung secara periodik dengan bimbingan seorang *mentor* dengan menggunakan pola pendekatan teman sebaya. Tujuan diadakannya mentoring tersebut yaitu terbentuknya Mahasiswa atau Mahasiswi muslim yang Islami.

2.3.2 Tujuan Kegiatan Mentoring

Dalam konteks kegiatan mentoring dalam lingkungan pendidikan dalam hal ini lingkungan kampus, tujuan diadakan kegiatan mentoring ini adalah lebih merujuk kepada pembinaan akhlaq yang dibimbing oleh beberapa orang yang telah berkompeten di bidangnya dan telah mendapatkan izin resmi dari pihak kampus baik Dekan dan Dosen pengampu mata kuliah Agama di tingkat Universitas. Dengan harapan adanya perbaikan- perbaikan baik dari sisi rohani dan moral pada diri masing-masing Mahasiswa dan Mahasiswi.

Rusmiyati (2003) menjelaskan mengenai tujuan mentoring agama Islam, yaitu:

- a) Mengajak para peserta mentoring untuk lebih mengenal dan mencintai Islam melalui kegiatan yang kreatif.
- b) Mengajak peserta mentoring untuk dapat aktif membaca Al-Quran.
- c) Memunculkan pemahaman yang benar terhadap Islam.
- d) Menangkal gerakan-gerakan yang bertujuan merusak moral generasi muda.

Setelah dijelaskan mengenai definisi dan tujuan mentoring agama Islam. Pada bagian selanjutnya akan dijelaskan metode dan bahan mentoring agama Islam.

2.3.3 Metode dalam Mentoring Agama Islam

Dalam sebuah aktivitas mentoring, dibutuhkan metode-metode pendekatan dan alokasi waktu yang sesuai untuk kegiatan tersebut. Hal ini berpengaruh terhadap daya konsentrasi dan *mood* para peserta mentoring.

Metode pendekatan yang digunakan adalah ceramah, *game* atau permainan, dan diskusi. Dalam penyampaian, setiap materi membutuhkan waktu 60 menit (satu jam), dengan alokasi waktu yang akan dibahas pada tabel di bawah ini, yaitu sebagai berikut (AAI FTI UII, 2015):

Tabel 2.1 Tabel Alokasi Waktu Metode Mentoring Agama Islam

Langkah	Uraian	Waktu
Pembukaan	Mentor membuka pertemuan dan menyampaikan tujuan materi	5 menit
Game (Optional)	Mentor melakukan simulasi sebagai pengantar materi	10 menit
Ceramah	Mentor menyampaikan isi materi	30 menit
Diskusi	Berdiskusi dan tanya jawab	10 menit
Penutup	Mentor menyimpulkan isi materi dan menutup pertemuan dengan do'a	5 menit

Selain itu, strategi atau metode untuk mentoring juga dijelaskan oleh Ruswandi (2004) yang tercakup dalam *motto* mentoringnya yaitu *fun, fresh, and focus*. *Fun* di sini adalah metode pengajaran dengan cara yang menyenangkan dan mengasikan seperti dengan cara menggunakan permainan sebagai alat pendukung bantu ajar. Kemudian *fresh, fresh* yaitu metode belajar dengan mengunjungi alam terbuka. Penggunaan metode ini peserta mentoring diajak untuk lebih dekat dengan alam terbuka sehingga peserta dapat melihat apa saja keagungan ciptaan tuhan.

Metode yang terakhir yaitu *focus, focus* adalah metode dengan mengadakan diskusi kelompok. Dengan diskusi tersebut, peserta dapat memfokuskan pikiran mereka dalam membahas materi yang dibahas pada kegiatan mentoring tersebut. Dari pengaplikasian ketiga metode kegiatan tersebut diharapkan proses kegiatan mentoring bisa berjalan dengan maksimal dan menyenangkan tanpa mengurangi nilai keIslaman di dalamnya.

2.4 *Monitoring*

Monitoring merupakan kegiatan yang mencakup aktifitas pelaporan dan peninjauan ulang suatu kegiatan yang dilakukan untuk tujuan tertentu, untuk memeriksa apakah kegiatan yang telah berjalan sesuai dengan sasaran dan tujuan (Mercy, 2005).

Pada dasarnya, *monitoring* memiliki dua fungsi dasar yang berhubungan, yaitu *compliance monitoring* dan *performance monitoring* (Mercy, 2005). *Compliance monitoring* berfungsi untuk memastikan proses sesuai dengan harapan atau rencana. Sedangkan, *performance monitoring* berfungsi untuk mengetahui perkembangan organisasi dalam pencapaian target yang diharapkan.

Umumnya, *output monitoring* berupa *progress report procces*. *Output* tersebut diukur secara deskriptif maupun non-deskriptif. *Output monitoring* bertujuan untuk mengetahui kesesuaian proses yang telah berjalan. *Output monitoring* berguna pada perbaikan mekanisme proses atau kegiatan dimana *monitoring* dilakukan.

Dalam konteks AAI ini, proses *monitoring* diaplikasikan sebagai pemantauan *report* kegiatan secara berkala dari kegiatan rutin AAI dari masing-masing asisten kepada dosen pembimbing. Dengan hasil *report* tersebut diharapkan proses kegiatan mentoring dapat diketahui perkembangan kemajuan kegiatan AAI pada tiap tahunnya. Sehingga dapat dijadikan bahan evaluasi dan perbaikan untuk kegiatan tahap selanjutnya.

2.5 Asistensi Agama Islam FTI

Asistensi Agama Islam (AAI) FTI merupakan salah satu program kerja keagamaan bidang 3 yang dirancang oleh Direktorat Pendidikan dan Pengembangan Agama Islam UII (DPPAI UII). Program ini kemudian diimplementasikan di beberapa fakultas, termasuk Fakultas Teknologi Industri UII (FTI UII). Program AAI ini mulai diimplementasikan di FTI pada tahun 2010 saat kepengurusan bapak Ir.Gumbolo Hadi Susanto, M.Sc, selaku Dekan FTI UII. Misi FTI UII yaitu mewujudkan institusi yang menjadi bagian dari dakwah Islamiyah sesuai dengan tuntunan Al-Quran dan Al-Hadits di bidang pendidikan, ilmu dan teknologi dan tempat dihasilkannya insan Ulil Albab dan mampu menunjukkan jati diri sebagai Khairo Ummah. Sesuai dengan misi tersebut program ini diharapkan dapat mendukung program keagamaan pada tingkat fakultas (Susanto, 2015).

Tujuan dibentuknya program mentoring AAI FTI ini adalah sebagai program khusus pendampingan matakuliah Agama Islam bagi mahasiswa baru pada tiap awal semester. Dengan adanya mentoring tersebut diharapkan mahasiswa dan mahasiswi baru FTI mendapatkan bimbingan mengenai ilmu agama dan praktek secara langsung dengan diampu oleh asisten AAI di luar jam kuliah.

Program mentoring AAI ini didukung oleh *stakeholder* yang terkait, yaitu Dekanat FTI, Lembaga Dakwah Fakultas CENTRIS FTI, Dosen Agama dan DPPAI. Peran dari masing-masing *stakeholder* tersebut adalah sebagai pendukung dalam perencanaan pelaksanaan evaluasi serta pengawasan dalam kelancaran pelaksanaan program mentoring AAI FTI UII.

Program kegiatan Asistensi Agama Islam ini diharapkan menjadi salah satu program rutin unggulan di setiap tahun ajaran berikutnya, serta diharapkan memberikan efek positif dan nilai *plus* bagi Fakultas dan juga Universitas.

2.6 Pengujian Perangkat Lunak

Pengujian perangkat lunak adalah proses pemeriksaan atau evaluasi sistem atau komponen sistem secara manual atau otomatis untuk memverifikasi apakah sistem memenuhi kebutuhan-kebutuhan yang dispesifikasikan atau mengidentifikasi perbedaan-perbedaan anatar hasil yang diharapkan dengan hasil yang terjadi (Hariyanto, 2004).

Manfaat pengujian (Hariyanto, 2004) antara lain sebagai berikut:

- a. Pengujian akan menyingkap kesalahan di perangkat lunak.
- b. Pengujian mendemonstrasikan fungsi-fungsi perangkat lunak bekerja sesuai spesifikasi, kebutuhan serta terpenuhi perilaku dan sejenisnya.

2.6.1 Teknik Pengujian

Teknik yang digunakan dalam pengujian *prototype* sistem *monitoring* mentoring ini yaitu Pengujian *Alpha* dengan metode *black box testing*. Penjelasan dari teknik pengujian tersebut akan dijelaskan secara rinci sebagai berikut:

2.6.1.1 Pengujian Alpha

Pengujian *Alpha* merupakan pengujian fungsionalitas yang digunakan untuk menguji sistem yang telah dibuat. Pengujian ini dilakukan terhadap perangkat lunak untuk memastikan bahwa perangkat lunak tersebut dapat berjalan dengan benar sesuai kebutuhan serta tujuan yang diharapkan. Teknik atau metode yang digunakan dalam pengujian *alpha* yaitu dengan metode *black-box testing*.

Black Box testing merupakan teknik pengujian untuk mendapatkan hasil *input* yang akan dijalankan semua persyaratan fungsional pada suatu program (Pressman, 2002). Pengujian ini difokuskan pada pengujian fungsionalitas dan *output* yang dihasilkan aplikasi. Menurut Black (2009 : 3), *tester* menggunakan *black-box test* untuk menemukan *bug* dalam *high level operations*, pada tingkatan

fitur, dan *scenario customer*. *Tester* dapat membuat pengujian fungsional *black-box* berdasarkan pada apa yang harus sistem lakukan.

Kategori yang terdapat pada pengujian *black box* yaitu sebagai berikut :

- a) Fungsi-fungsi yang salah atau hilang
- b) Kesalahan *Interface* (antarmuka)
- c) Kesalahan kinerja dalam sistem.

