

MAKALAH TUGAS AKHIR

APLIKASI *E-LEARNING* MENTORING AGAMA ISLAM FAKULTAS TEKNIK UNIVERSITAS DIPONEGORO

Muhammad Guruh Wicaksana (L2F 001 621)

Jurusan Teknik Elektro, Fakultas Teknik, Universitas Diponegoro

ABSTRAK

Teknologi informasi berkembang sangat cepat dan makin diperkuat keberadaannya dengan adanya internet yang sudah tersebar diseluruh dunia, sehingga hampir semua masyarakat dapat menggunakan internet. Dengan adanya internet arus informasi makin cepat berputar kepada masyarakat dan aplikasi-aplikasi teknologi informasi mulai bermunculan. Aplikasi-aplikasi tersebut memberikan kemudahan bagi masyarakat yang menggunakannya, diantaranya e-mail, video conference, mailing list, chatting, e-learning dan lainnya.

Tugas akhir ini mengimplementasikan sistem e-learning pada mentoring agama Islam di Fakultas Teknik Universitas Diponegoro. Sistem e-learning sangat potensial untuk membuat proses mentoring lebih efektif, sebab peluang mentee untuk berinteraksi dengan pementor, antarmenit, maupun bahan belajarnya terbuka lebih luas. Mentee dapat berkomunikasi dengan pementornya kapan saja melalui internet. Melalui e-learning, para mentee dimungkinkan untuk tetap dapat belajar sekalipun tidak hadir secara fisik di dalam pertemuan mentoring mingguan. Kegiatan mentoring menjadi sangat fleksibel karena dapat disesuaikan dengan ketersediaan waktu para mentee. Kegiatan pembelajaran terjadi melalui interaksi mentee dengan sumber belajar yang tersedia dalam aplikasi e-learning dan dapat diakses dari internet.

Aplikasi e-learning ini dapat digunakan sebagai fasilitas penunjang dari mentoring agama Islam yang menggunakan cara konvensional dimana peserta mentoring bertemu tiap minggunya dengan pementor masing-masing. Dengan aplikasi e-learning pada mentoring agama Islam, mentee dapat berinteraksi dengan pementornya dimanapun dan kapanpun mereka berada. Dengan prasyarat mereka terkoneksi dengan internet.

Kata kunci : e-learning, teknologi informasi, internet, mentoring agama Islam

I. PENDAHULUAN

1.1 Latar Belakang

Mentoring agama Islam adalah suatu proses pendidikan yang harus dilalui setiap mahasiswa muslim yang mengambil mata kuliah agama Islam. Pengelolaan mentoring agama Islam ini dilakukan oleh Unit Kegiatan Mahasiswa (UKM) Rohani Islam dari tingkat Universitas hingga tingkat Jurusan melalui koordinasi langsung dengan seluruh dosen agama Islam.

Dengan berkembangnya teknologi informasi muncul pemikiran untuk menjalankan metode mentoring agama Islam ini secara virtual atau maya melalui internet. Seperti telah diketahui bahwa internet bukanlah hal baru untuk mahasiswa dan cukup mudah untuk ditemukan. Konsep ini menawarkan cara belajar yang berbeda dari cara konvensional. Peranan pementor atau pengajar menjadi tidak begitu dominan lagi. Peserta mentoring memiliki peranan yang lebih dominan. Dalam konsep ini peserta mentoring dituntut untuk belajar sendiri dari materi-materi yang diberikan pementor. Walaupun peran peserta mentoring lebih dominan, tetap ada seorang pementor yang memandu proses mentoring. Keaktifan peserta mentoring terutama dalam diskusi akan menjadikan cara berpikir semakin kritis. Hal-hal tersebut mendukung terciptanya kualitas pribadi muslim yang baik.

Manfaat yang didapat dari aplikasi ini salah satunya adalah peserta mentoring dapat melakukan proses mentoring dimana pun ia berada selama ada fasilitas internet di wilayah ia berada. Bagi mahasiswa muslim,

sarana ini dapat menjadi penunjang peningkatan pengetahuan Islam dengan sedikitnya waktu kuliah agama Islam dalam sistem pendidikan Universitas. Aplikasi ini tidak bermaksud untuk mengubah bentuk dari mentoring konvensional yang sudah ada, namun aplikasi ini dapat menjadi pilihan metode baru dengan makin berkembangnya teknologi informasi dan internet.

1.2 Tujuan dan Manfaat

Tujuan yang akan dicapai dalam tugas akhir ini adalah merancang dan membangun aplikasi *e-learning* yang diimplementasikan pada mentoring agama Islam Fakultas Teknik Universitas Diponegoro dengan berbasis web, serta merancang dan membangun perangkat-perangkat pendukung yang dibutuhkan dalam pelaksanaan mentoring agama Islam berbasis web. Sehingga aplikasi ini dapat menjadi penunjang pembelajaran mentoring agama Islam di Fakultas Teknik Universitas Diponegoro.

1.3 Batasan Masalah

1. Aplikasi yang dibangun merupakan aplikasi *e-learning* dengan berbasis web.
2. Bahasa pemrograman yang digunakan adalah PHP dan MySQL.
3. Sistem dibangun menggunakan server Apache dengan tingkat keamanan standar yaitu menggunakan *session*.

II. SISTEM *E-LEARNING* MENTORING AGAMA ISLAM

2.1 Definisi *E-learning*

Ilustrasi berikut ini mungkin akan dapat membantu memperjelas pengertian tentang *e-learning* (Newsletter of ODLQC, 2001). “Ada seseorang yang membawa laptop ke sebuah tempat yang berada jauh di gugusan kepulauan kecil yang terpencil. Dari tempat yang sangat terpencil ini, orang tersebut mulai menggunakan laptop-nya dan melakukan akses terhadap berbagai materi program pelatihan yang tersedia. Tidak ada layanan bantuan belajar dari tutor maupun dukungan layanan belajar bentuk lainnya. Dalam konteks ini, apakah orang tersebut dapat dikatakan telah melaksanakan *e-learning*? Jawabannya adalah TIDAK. Mengapa? Karena yang bersangkutan di dalam kegiatan pembelajaran yang dilakukannya tidak memperoleh layanan bantuan belajar dari tutor maupun layanan bantuan belajar lainnya. Bagaimana kalau yang bersangkutan mempunyai telepon genggam dan kemudian berhasil menggunakannya untuk menghubungi seorang tutor? Apakah dalam konteks yang demikian ini dapat dikatakan bahwa yang bersangkutan telah melaksanakan *e-learning*? Jawabannya adalah YA.”

Dari ilustrasi tersebut di atas, setidaknya-tidaknya dapat ditarik 3 (tiga) hal penting sebagai persyaratan kegiatan belajar elektronik (*e-learning*), yaitu:

- (a) Kegiatan pembelajaran dilakukan melalui pemanfaatan jaringan (“jaringan” dalam uraian ini dibatasi pada penggunaan internet. Jaringan dapat saja mencakup LAN atau WAN)
- (b) Tersedianya dukungan layanan belajar yang dapat dimanfaatkan oleh peserta belajar, misalnya CD-ROM, atau bahan cetak
- (c) Tersedianya dukungan layanan tutor yang dapat membantu peserta belajar apabila mengalami kesulitan.

Dengan demikian, secara sederhana dapat dikatakan bahwa pembelajaran elektronik (*e-learning*) merupakan kegiatan pembelajaran yang memanfaatkan jaringan (Internet, LAN, WAN) sebagai metode penyampaian, interaksi, dan fasilitasi serta didukung oleh berbagai bentuk layanan belajar lainnya (Brown, 2000; Feasey, 2001).

2.2 Fungsi *E-learning*

Setidaknya ada 3 (tiga) fungsi pembelajaran elektronik terhadap kegiatan pembelajaran di dalam kelas (*classroom instruction*), yaitu :

1. Suplemen (Tambahan)

Dikatakan berfungsi sebagai suplemen (tambahan), apabila peserta didik mempunyai kebebasan memilih, apakah akan memanfaatkan materi pembelajaran elektronik atau tidak. Dalam hal ini, tidak ada kewajiban/keharusan bagi peserta didik untuk mengakses materi pembelajaran elektronik. Sekalipun sifatnya opsional, peserta didik yang memanfaatkannya

tentu akan memiliki tambahan pengetahuan atau wawasan.

2. Komplemen (Pelengkap)

Dikatakan berfungsi sebagai komplemen (pelengkap) apabila materi pembelajaran elektronik diprogramkan untuk melengkapi materi pembelajaran yang diterima siswa di dalam kelas (Lewis, 2002). Materi pembelajaran elektronik dikatakan sebagai *enrichment*, apabila kepada peserta didik yang dapat dengan cepat menguasai/memahami materi pelajaran yang disampaikan guru secara tatap muka (*fast learners*) diberikan kesempatan untuk mengakses materi pembelajaran elektronik yang memang secara khusus dikembangkan untuk mereka. Tujuannya agar semakin memantapkan tingkat penguasaan peserta didik terhadap materi pelajaran yang disajikan guru di dalam kelas. Dikatakan sebagai program *remedial*, apabila kepada peserta didik yang mengalami kesulitan memahami materi pelajaran yang disajikan guru secara tatap muka di kelas (*slow learners*) diberikan kesempatan untuk memanfaatkan materi pembelajaran elektronik yang memang secara khusus dirancang untuk mereka. Tujuannya agar peserta didik semakin lebih mudah memahami materi pelajaran yang disajikan guru di kelas.

3. Substitusi (Pengganti)

Beberapa perguruan tinggi di negara-negara maju memberikan beberapa alternatif model kegiatan pembelajaran/perkuliahan kepada para mahasiswanya. Tujuannya agar para mahasiswa dapat secara fleksibel mengelola kegiatan perkuliahannya sesuai dengan waktu dan aktivitas lain sehari-hari mahasiswa. Ada 3 alternatif model kegiatan pembelajaran yang dapat dipilih peserta didik, yaitu: (1) sepenuhnya secara tatap muka (konvensional), (2) sebagian secara tatap muka dan sebagian lagi melalui internet, atau bahkan (3) sepenuhnya melalui internet.

2.3 Manfaat *E-learning*

Manfaat pembelajaran elektronik menurut A. W. Bates (Bates, 1995) dan K. Wulf (Wulf, 1996) terdiri atas 4 hal, yaitu:

- (1) Meningkatkan kadar interaksi pembelajaran antara peserta didik dengan guru atau instruktur (*enhance interactivity*).
- (2) Memungkinkan terjadinya interaksi pembelajaran dari mana dan kapan saja (*time and place flexibility*).
- (3) Menjangkau peserta didik dalam cakupan yang luas (*potential to reach a global audience*).
- (4) Mempermudah penyempurnaan dan penyimpanan materi pembelajaran (*easy updating of content as well as archivable capabilities*) dengan pengelolaan kegiatan pembelajarannya sendiri. Harus ada komitmen dari guru/dosen/instruktur yang akan memantau perkembangan kegiatan belajar peserta didiknya dan sekaligus secara teratur memotivasi peserta didiknya.

2.4 Mentoring Agama Islam

Mentoring agama Islam adalah pendampingan dalam mata kuliah agama Islam yang dikelola oleh departemen mentoring Kerohanian Islam (Rohis). Rohis bekerjasama dengan dosen mata kuliah agama Islam untuk melaksanakan mentoring agama Islam.

Setiap mahasiswa muslim yang mengambil mata kuliah agama Islam yang pada umumnya berada pada semester satu harus mengikuti mentoring agama Islam yang dikelola oleh Rohis Jurusan melalui koordinasi dengan Rohis Fakultas. Proses mentoring berjalan selama satu semester, setelah satu semester mahasiswa muslim dapat mengikuti *follow-up* mentoring.

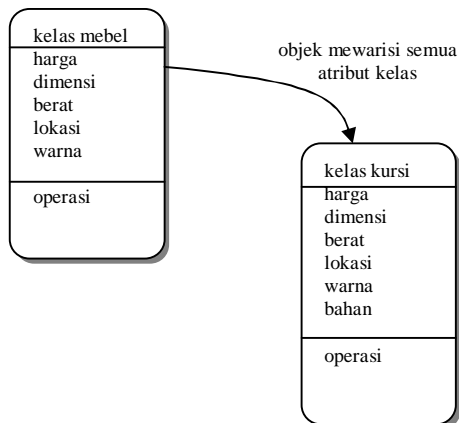
Mentoring agama Islam dilakukan satu minggu sekali dengan waktu yang telah disepakati dalam kelompok tersebut. Proses jalannya mentoring agama Islam diawali dengan adanya pembukaan mentoring agama Islam. Pada acara tersebut, setiap mahasiswa muslim akan dibagi menjadi beberapa kelompok oleh departemen mentoring yang kemudian akan didampingi oleh satu orang pementor.

III. METODE PERANCANGAN SISTEM

BERORIENTASI OBJEK

3.1 Konsep Berorientasi Objek

Konsep berorientasi objek merupakan salah satu konsep dalam pembangunan atau rekayasa perangkat lunak. Untuk memahami konsep berorientasi objek, perlu terlebih dahulu memahami istilah “berorientasi objek” dan “objek”. Dalam dunia nyata, dapat kita ambil contoh kursi. Kursi adalah anggota dari kelas objek yang lebih luas yaitu mebel. Serangkaian atribut generik dapat diasosiasikan dengan setiap objek pada kelas mebel. Misalnya, semua mebel memiliki harga, dimensi, berat, lokasi, dan warna. Atribut-atribut tersebut dapat diberlakukan terhadap setiap anggota kelas mebel misalnya meja, kursi, sofa, atau lemari. Karena kursi adalah anggota dari kelas mebel, maka kursi mewarisi semua atribut yang ditentukan untuk kelas tersebut. Gambar 3.1 menunjukkan hubungan antara mebel dan kursi secara skematis.

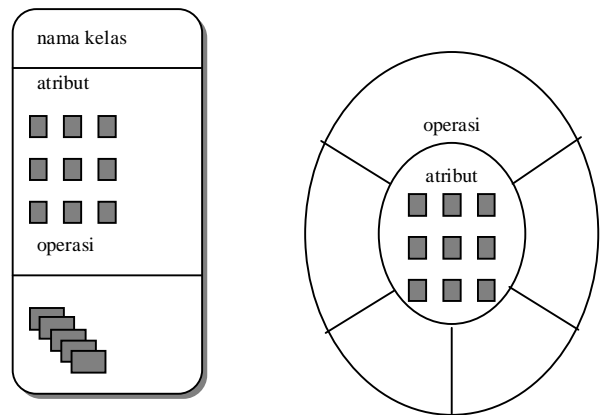


Gambar 3.1 Pewarisan dari kelas ke objek

Objek kursi (dan semua objek lain secara umum) mengenkapsulasi data (nilai atribut yang menentukan kursi), operasi (aksi yang diaplikasikan untuk mengubah-ubah atribut kursi), dan objek lain. Enkapsulasi berarti bahwa semua informasi itu dikemas di bawah satu nama dan dapat digunakan lagi sebagai satu spesifikasi atau komponen program.

3.2 Kelas dan Objek

Kelas adalah konsep berorientasi objek (*Object Oriented*, OO) yang terdiri atas abstraksi data dan prosedural yang diperlukan untuk menggambarkan isi serta tingkah laku berbagai entitas atau objek dalam dunia nyata. Sebuah kelas memiliki atribut atau abstraksi data yang menggambarkan kelas. Di samping itu, sebuah kelas juga memiliki abstraksi prosedural yang disebut operasi atau metode. Gambar 2 menggambarkan notasi suatu kelas.



Gambar 3.2 Representasi dari kelas berorientasi objek

3.3 Atribut

Pada setiap kelas dan objek terdapat atribut yang menggambarkan kelas atau objek tersebut. Hubungan biner mengimplikasikan bahwa atribut dapat menerima nilai yang ditentukan oleh domain yang ditambahkan. Pada kebanyakan kasus, domain adalah serangkaian nilai spesifik. Sebagai contoh, anggap bahwa kelas mobil memiliki atribut warna. Domain dari nilai untuk warna adalah {putih, hitam, perak, abu-abu, biru, merah, kuning, hijau}. Dalam situasi yang lebih kompleks, domain tersebut dapat merupakan himpunan kelas. Kelas mobil juga memiliki atribut *power train* yang meliputi nilai domain sebagai berikut: {pilihan ekonomi 16 *valve*, pilihan sport 24 *valve*, pilihan mewah 32 *valve*}. Masing-masing “pilihan” memiliki serangkaian atribut spesifiknya sendiri-sendiri.

3.4 Operasi atau metode

Objek mengenkapsulasi data (direpresentasikan sebagai sekumpulan atribut) dan algoritma yang memproses data. Algoritma-algoritma ini disebut dengan operasi atau metode dan dapat dipandang sebagai modul dalam kerangka fungsional.

Masing-masing operasi yang dienkapsulasi oleh suatu objek memberikan representasi mengenai satu dari tingkah laku objek. Sebagai contoh, operasi *GetColor* untuk objek mobil akan mengekstrak warna yang disimpan pada atribut warna. Implikasi dari operasi itu adalah bahwa kelas mobil telah didesain untuk menerima stimulus (pesan) yang membutuhkan warna dari keadaan kelas tertentu. Setiap kali objek menerima suatu stimulus objek tersebut menginisiasi berbagai macam tingkah laku. Hal ini dapat sesederhana pemanggilan kembali warna mobil atau serumit inisiasi rantai stimulus yang dilewatkan di antara objek yang berbeda

3.5 Pesan

Pesan adalah perantara dimana objek-objek berinteraksi. Pesan menstimulasi banyak tingkah laku yang ditemui di dalam objek penerimaan. Tingkah laku dikerjakan bila suatu operasi dieksekusi.

3.6 Enkapsulasi dan Pewarisan

Gambar 3.2 menunjukkan bahwa atribut yang menggambarkan kelas diikuti oleh sebuah “dinding” abstraksi prosedural (yang disebut operasi atau metode) yang mampu memanipulasi data dengan berbagai cara. Satu-satunya cara untuk mencapai atribut itu (dan mengoperasikannya) adalah dengan melalui salah satu dari metode-metode yang membangun dinding itu. Dengan demikian, kelas tersebut mengenkapsulasi data dan pemrosesan yang membentuk kelas tersebut.

Pewarisan adalah satu dari pembeda kunci di antara sistem OO dan konvensional. Sebagai contoh subkelas Y mewarisi semua atribut dan operasi yang sesuai dengan superkelas X. Itu berarti bahwa semua struktur data dan algoritma yang semula didesain dan diimplementasikan untuk X segera dapat diperoleh untuk Y tanpa melakukan kerja selanjutnya. Penggunaan ulang dilakukan secara langsung.

3.7 Analisis Berorientasi Objek

Analisis berorientasi objek (*Object Oriented Analysis*, OOA) merupakan aktivitas teknik pertama yang dilakukan sebagai bagian dari rekayasa perangkat lunak berorientasi objek. OOA didasarkan pada serangkaian prinsip dasar, yaitu:

- (1) domain informasi dimodelkan;
- (2) fungsi model digambarkan;
- (3) tingkah laku model direpresentasikan;
- (4) model dipartisi untuk mengekspos detail yang lebih besar;
- (5) model awal merepresentasikan inti masalah sementara model selanjutnya memberikan detail implementasi.

Tujuan OOA adalah menentukan semua kelas (dan hubungan serta tingkah laku yang berkaitan dengannya) yang relevan dengan masalah yang akan dipecahkan. Untuk melakukannya, sejumlah tugas harus dilakukan :

1. Persyaratan pemakai dasar harus dikomunikasikan diantara pelanggan dan perencana perangkat lunak.
2. Kelas-kelas harus diidentifikasi (misalnya, atribut dan metode yang ditentukan).
3. Hirarki kelas harus dispesifikasikan.
4. Hubungan objek ke objek (koneksi objek) harus direpresentasikan.
5. Tingkah laku objek dimodelkan.
6. Tugas 1 sampai 5 harus diaplikasikan lagi secara iteratif sampai model selesai.

3.8 Desain Berorientasi Objek

Desain berorientasi objek (*Object Oriented Design*, OOD) mentransformasi model analisis yang dibuat dengan menggunakan OOA ke dalam suatu model desain yang berfungsi sebagai rancang bangun perangkat lunak. Gambar 3.3 merupakan piramida desain OO yang terdiri dari empat lapisan sebagai berikut:

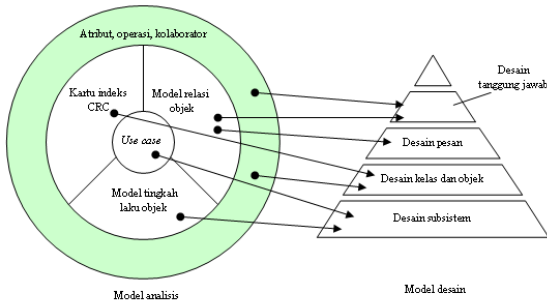
1. **Lapisan subsistem**, berisi representasi masing-masing subsistem yang memungkinkan perangkat lunak mencapai persyaratan yang didefinisikan oleh pelanggannya dan untuk mengimplementasi infrastruktur teknik yang mendukung persyaratan pelanggan.
2. **Lapisan objek dan kelas**, berisi hirarki kelas yang memungkinkan sistem diciptakan dengan menggunakan generalisasi dan spesialisasi yang ditarget secara perlahan. Lapisan ini juga berisi infrastruktur yang mendukung persyaratan pelanggan.
3. **Lapisan pesan**, berisi detail yang memungkinkan masing-masing objek berkomunikasi dengan kolaboratornya. Lapisan ini membangun antarmuka internal dan eksternal bagi sistem tersebut.
4. **Lapisan tanggung jawab**, berisi struktur desain data dan algoritma untuk semua atribut dan operasi masing-masing objek.



Gambar 3.3 Piramida desain OO

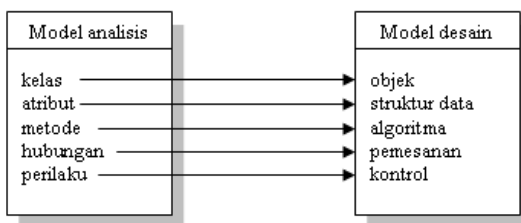
Gambar 3.4 menunjukkan translasi dari model analisis OO ke model desain OO. Desain subsistem diperoleh dengan mempertimbangkan keseluruhan persyaratan pelanggan (direpresentasikan dengan *use case*) dan kejadian serta keadaan yang dapat diamati

secara eksternal (model tingkah laku objek). Kelas dan desain objek dipetakan dari deskripsi atribut, operasi, dan kolaborasi yang diisikan pada model CRC. Desain pesan dikendalikan oleh model hubungan objek, dan desain tanggung jawab ditarik dengan menggunakan atribut, operasi, dan kolaborasi yang digambarkan pada model CRC.



Gambar 3.4 Translasi model OOA ke dalam model OOD

Seperti diperlihatkan pada Gambar 3.4, dari model OOA ke model OOD terjadi translasi. Gambar 3.5 berikut menggambarkan translasi dari setiap komponen model analisis ke model desain.



Gambar 3.5 Translasi model analisis ke dalam model desain selama desain objek

3.9 Proses OOA

Proses OOA tidak dimulai dengan suatu pemikiran melalui objek, melainkan dengan memahami cara sistem akan digunakan oleh manusia (bila sistem adalah *human-interactive*), oleh mesin (bila sistem melibatkan di dalam kontrol proses), atau oleh program yang lain (bila sistem berkoordinasi dan mengontrol aplikasi). Untuk itulah didefinisikan skenario kegunaan (*use case*) yang menggambarkan interaksi sistem dan kegunaan sistem.

a. Use Case

Pengumpulan persyaratan selalu merupakan langkah pertama pada setiap aktivitas analisis perangkat lunak. Berdasarkan persyaratan tersebut, perancang perangkat lunak (analisis) dapat menciptakan serangkaian skenario yang masing-masing mengidentifikasi urutan pemakaian dari sistem yang akan dibangun. Skenario tersebut, yang sering disebut dengan *use case*, memberikan deskripsi mengenai bagaimana sistem akan digunakan.

Untuk membuat *use case*, analisis terlebih dahulu mengidentifikasi tipe manusia (atau perangkat) yang menggunakan sistem atau produk tersebut. Aktor-aktor

tersebut merepresentasikan peran yang dimainkan oleh manusia atau perangkat pada saat sistem beroperasi. Bila didefinisikan secara lebih formal, aktor adalah sesuatu yang berkomunikasi dengan sistem atau produk dan bersifat eksternal terhadap sistem itu sendiri.

Setelah aktor-aktor diidentifikasi *use case* dapat dikembangkan. *Use case* menggambarkan cara di mana aktor berinteraksi dengan sistem. Isi *use case* harus dapat menjawab pertanyaan-pertanyaan sebagai berikut:

1. Apakah tugas atau fungsi utama yang dilakukan oleh aktor ?
2. Sistem informasi apakah yang akan didapat atau diperoleh, dihasilkan, atau diubah oleh aktor ?
3. Haruskah aktor memberi tahu sistem mengenai perubahan di dalam lingkungan eksternal ?
4. Informasi apa yang diinginkan oleh aktor dari sistem tersebut ?
5. Apakah aktor ingin diberi tahu mengenai perubahan-perubahan yang tidak diharapkan ?

b. Pemodelan Kelas-Tanggung Jawab-Kolaborator (CRC)

Setelah skenario pemakaian dasar dikembangkan untuk sistem, maka saat itulah waktunya untuk mengidentifikasi kelas-kelas calon, dan menunjukkan tanggung jawab serta kolaborasi mereka. Pemodelan CRC memberikan cara sederhana untuk mengidentifikasi dan mengumpulkan kelas-kelas yang relevan dengan persyaratan produk atau sistem.

Pada dasarnya, model CRC dapat menggunakan kartu indeks virtual dan aktual. Maksudnya adalah untuk mengembangkan representasi kelas yang terorganisasi. Tanggung jawab adalah atribut dan operasi yang relevan untuk kelas. Pendeknya, tanggung jawab adalah “sesuatu yang diketahui atau dilakukan oleh kelas”. Kolaborator adalah kelas-kelas lain yang diperlukan untuk memberikan informasi yang dibutuhkan oleh suatu kelas untuk menyelesaikan tanggung jawab. Secara umum, kolaborasi mengimplikasikan permintaan informasi atau permintaan beberapa aksi. Gambar 3.6 merupakan model dari CRC yang berupa kartu indeks.

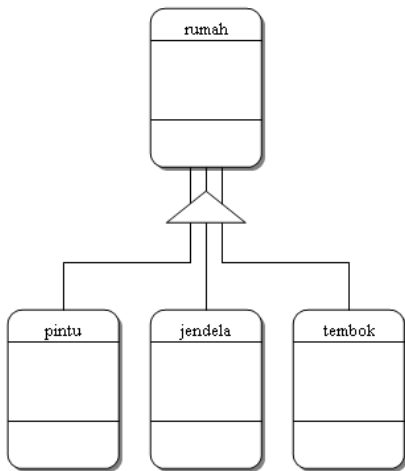
nama kelas	
tipe kelas: (misal: perangkat, properti, peran, kejadian,...)	
karakteristik kelas: (misal: dapat dilihat, atomik, konkuren,...)	
tanggung jawab	kolaborasi

Gambar 3.6 Kartu indeks model CRC

c. Pendefinisian Struktur dan Hirarki

Setelah kelas dan objek didefinisi dengan menggunakan model CRC, fokus perhatian diarahkan pada penentuan struktur model kelas dan hirarki kelas yang muncul pada saat kelas dan subkelas muncul.

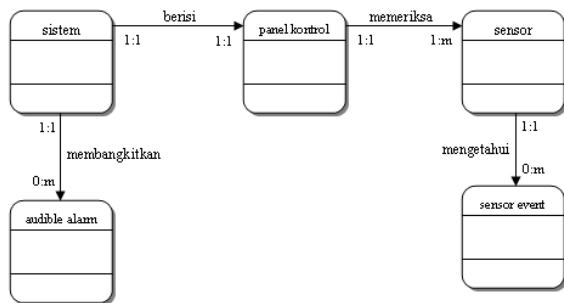
Gambar 3.7 memperlihatkan hirarki kelas sederhana dari objek rumah.



Gambar 3.7 Hirarki sederhana dari kelas rumah

d. Model Hubungan Objek

Hubungan dapat ditarik dengan menguji kata kerja atau frase kerja di dalam pernyataan ruang lingkup atau *use case* untuk sistem. Dengan menggunakan uraian gramatika, analisis mengisolasi kata kerja yang menunjukkan lokasi fisik atau penempatan, komunikasi, kepemilikan, dan pemenuhan suatu kondisi. Hal-hal tersebut memberikan indikasi suatu hubungan. Gambar 3.8 merupakan contoh hubungan antar-objek dalam sistem *safehome* dengan kardinalitas yang telah ditentukan.



Gambar 3.8 Hubungan antar-objek dalam sistem *safehome*

Model hubungan objek (seperti model hubungan entitas) dapat diperoleh dalam tiga langkah, yaitu:

1. Dengan menggunakan kartu indeks CRC, jaringan objek kolaborator dapat ditarik. Gambar 3.8 merepresentasikan koneksi kelas untuk objek *safehome*.
2. Dengan mengkaji kartu indeks model CRC, tanggung jawab dan kolaborator dievaluasi dan masing-masing garis dihubungkan dengan yang tidak berlabel dan diberi nama. Untuk menghindari ambiguitas, digunakan anak panah untuk menunjukkan arah hubungan tersebut.
3. Begitu hubungan yang diberi nama telah dibangun, maka masing-masing ujung dievaluasi untuk menentukan kardinalitas (Gambar 3.8). Empat

pilihan muncul, yaitu: 0 ke 1, 1 ke 1, 0 ke banyak, atau 1 ke banyak. Misalnya, sistem *safehome* berisi sebuah panel kontrol tunggal (notasi kardinalitas 1:1 menunjukkan ini). Paling tidak satu sensor harus ada untuk *polling* oleh panel kontrol. Tetapi mungkin juga ada beberapa sensor lain (notasi 1:m menunjukkan ini).

e. Model Tingkah Laku Objek

Model CRC dan model hubungan objek merepresentasikan elemen statis dari model analisis OO. Transisi ke tingkah laku dinamis dapat dilakukan dengan membuat model tingkah laku objek. Untuk melakukan hal tersebut diperlukan tingkah laku sistem sebagai fungsi dari kejadian dan waktu spesifik.

Model tingkah laku objek menunjukkan bagaimana sistem OO akan merespon kejadian atau stimulus eksternal. Untuk menciptakan model tersebut, diperlukan langkah-langkah sebagai berikut:

1. Mengevaluasi semua *use case* agar dapat memahami urutan interaksi di dalam sistem.
2. Mengidentifikasi kejadian yang mengendalikan urutan interaksi serta memahami bagaimana kejadian-kejadian tersebut berhubungan dengan objek spesifik.
3. Melakukan penelusuran kejadian untuk masing-masing *use case*.
4. Membangun diagram perubahan keadaan (*state transition*) untuk sistem.
5. Mengkaji model tingkah laku objek untuk memverifikasi akurasi dan konsistensi.

3.10 Metode Pengujian Sistem

Pengujian perangkat lunak merupakan proses eksekusi program atau perangkat lunak dengan tujuan mencari kesalahan atau kelemahan dari program tersebut. Proses tersebut dilakukan dengan mengevaluasi atribut dan kemampuan program. Suatu program yang diuji akan dievaluasi apakah keluaran atau output yang dihasilkan telah sesuai dengan yang diinginkan atau tidak. Ada berbagai macam metode pengujian, teknik *black box* dan teknik *white box* merupakan metode pengujian yang telah dikenal dan banyak digunakan oleh pengembang perangkat lunak.

a. Metode Pengujian Black Box

Metode pengujian *black box* merupakan metode pengujian dengan pendekatan yang mengasumsikan sebuah sistem perangkat lunak atau program sebagai sebuah kotak hitam (*black box*). Pendekatan ini hanya mengevaluasi program dari output atau hasil akhir yang dikeluarkan oleh program tersebut. Program dipandang sebagai suatu kotak hitam sehingga pengujian hanya dilakukan pada bagian luar atau input dan output program. Struktur program dan kode-kode yang ada di dalamnya tidak termasuk dalam pengujian ini. Keuntungan dari metode pengujian ini adalah murah dan sederhana. Namun, pengujian dengan metode ini

tidak dapat mendeteksi kekurangefektifan pengkodean dalam suatu program.

b. Metode Pengujian White Box

Metode pengujian *white box* atau dapat disebut juga *glass box* merupakan metode pengujian dengan pendekatan yang mengasumsikan sebuah perangkat lunak atau program sebagai kotak kaca (*glass box*). Program dipandang sebagai suatu kotak kaca sehingga struktur program dan kode-kode yang ada di dalamnya dapat dilihat dan dievaluasi. Pendekatan ini akan mengevaluasi stuktur program dan kodenya yang meliputi keefektifan pengkodean, pernyataan kondisional (alur program), dan *looping* yang digunakan dalam program. Keuntungan dari metode pengujian ini antara lain dapat ditemukannya kode-kode tersembunyi yang menghasilkan kesalahan serta dapat menghasilkan program yang efektif. Dengan evaluasi alur dan kode program, pengujian ini akan memakan waktu yang lama dan biaya yang lebih mahal.

IV. IMPLEMENTASI DAN PENGUJIAN SISTEM E-LEARNING

4.1 Pengguna Sistem E-learning

Pengguna dari sistem *e-learning* terdiri dari 5 jenis, yaitu :

1. Administrator, merupakan pemegang hak akses paling tinggi dalam sistem.
2. Departemen mentoring, memiliki peran yang penting dalam mengelola mentoring agama Islam
3. Pementor, merupakan pembimbing mentee dalam mentoring agama Islam
4. Mentee, adalah pengguna yang mendapatkan bimbingan dari pementor
5. Dosen, memiliki hak untuk melihat atau memantu jalannya mentoring agama Islam

4.2 Proses Belajar Mengajar

a. Sisi Pementor



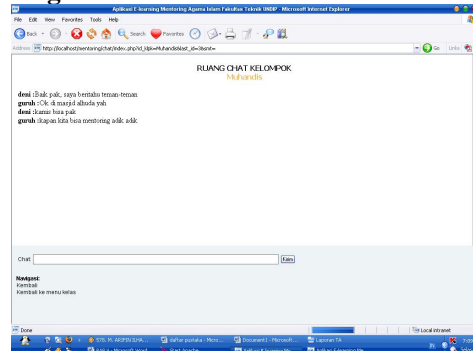
Gambar 4.1 Antarmuka sisi pementor

b. Sisi Mentee



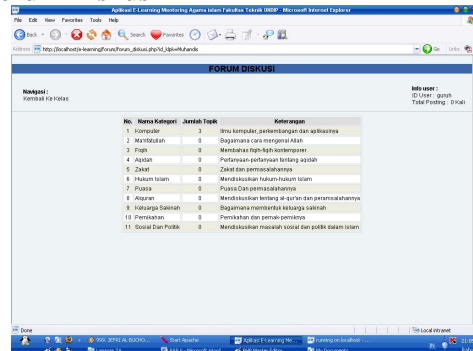
Gambar 4.2 Antarmuka sisi mentee

c. Ruang Chat



Gambar 4.3 Antarmuka ruang chat

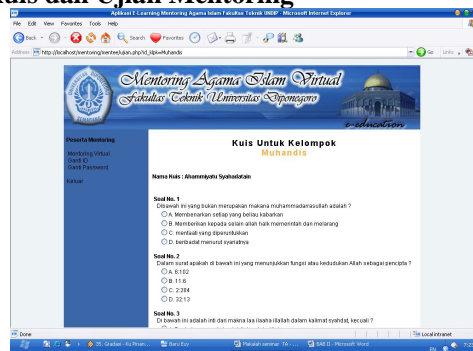
d. Forum Diskusi



Gambar 4.4 Antarmuka forum diskusi

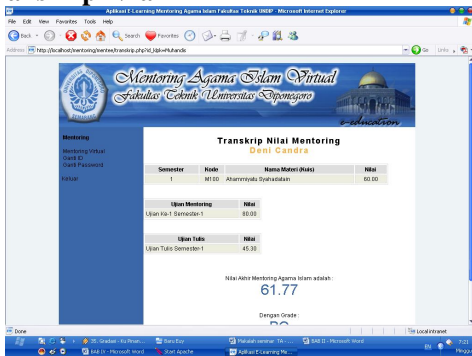
4.3 Evaluasi dan Pemantauan Kualitas

a. Kuis dan Ujian Mentoring



Gambar 4.5 Antarmuka pelaksanaan kuis/ujian mentoring

b. Transkrip Nilai

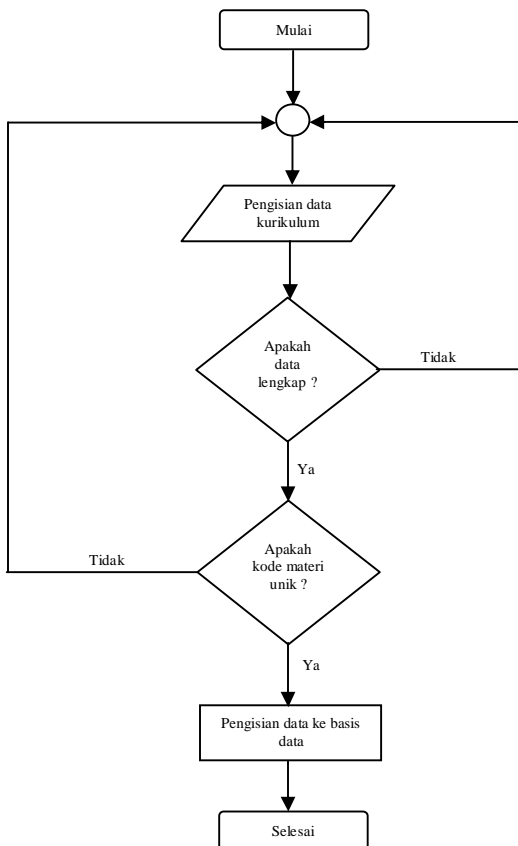


Gambar 4.6 Antarmuka transkrip nilai

4.4 Pengujian White Box

Metode pengujian *white box* atau dapat disebut juga *glass box* merupakan metode pengujian dengan pendekatan yang mengasumsikan sebuah perangkat lunak atau program sebagai kotak kaca (*glass box*).

Mengingat kompleksitas sistem, maka lingkup pembahasan hasil pengujian dalam makalah tugas akhir ini hanya pada penginputan kurikulum dan menampilkan tabel kurikulum. Pada tahapan persiapan sistem, departemen mentoring dapat membuat kurikulum dengan menambahkan materi mentoring pada setiap semester dan juga menampilkan tabel kurikulum.



Gambar 4.7 Diagram alir proses penginputan kurikulum

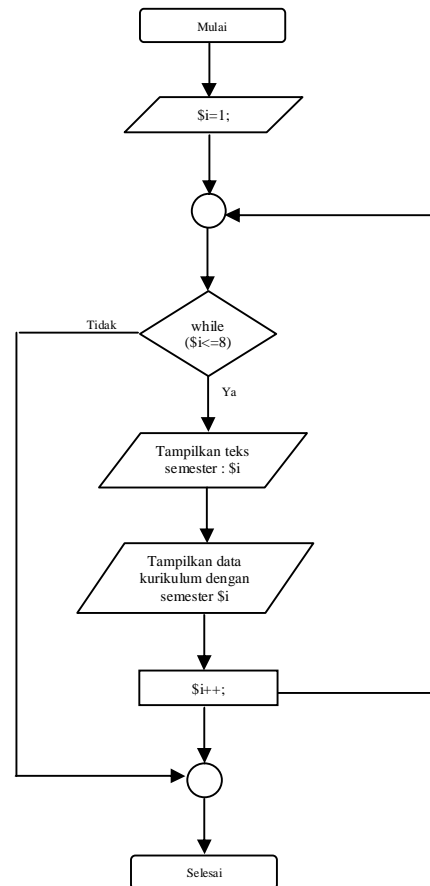
Tabel 4.1 Hasil pengujian *white box* pada proses penginputan kurikulum dengan perbandingan pernyataan kondisional

Kondisi 1	Kondisi 2	Output
FALSE	diabaikan	Error: input tidak lengkap
TRUE	FALSE	Error: kode materi telah digunakan
TRUE	TRUE	Input kurikulum sukses

Keterangan:

Kondisi 1 : if ((!\$kode_materi) or (!\$nama_materi) or (!\$smt))
//Kondisi jika input tidak lengkap

Kondisi 2 : if (\$cek) //Kondisi jika kode materi telah digunakan



Gambar 4.8 Diagram alir proses *looping* untuk menampilkan tabel kurikulum

Hasil pengujian *looping* untuk menampilkan daftar tabel kurikulum dapat dirangkum pada Tabel 4.2.

Tabel 4.2 Hasil pengujian *looping* untuk menampilkan tabel kurikulum

Input	Hasil yang diharapkan		Output hasil pengujian
	jumlah iterasi	output	
1*	8	Menampilkan daftar kurikulum semester ke-1	Menampilkan daftar kurikulum semester ke-1
2	7	Menampilkan daftar kurikulum	Menampilkan daftar kurikulum

		semester ke-2	semester ke-2							tidak lengkap	tidak lengkap
3	6	Menampilkan daftar kurikulum semester ke-3	Menampilkan daftar kurikulum semester ke-3	√	√	-	-	-	-	Error: Input tidak lengkap	Error: Input tidak lengkap
				-	√	√	-	-	-	Error: Input tidak lengkap	Error: Input tidak lengkap
				-	-	√	√	-	-	Error: Input tidak lengkap	Error: Input tidak lengkap
4	5	Menampilkan daftar kurikulum semester ke-4	Menampilkan daftar kurikulum semester ke-4	-	-	-	√	√	-	Error: Input tidak lengkap	Error: Input tidak lengkap
				√ ⁽¹⁾	√	√	-	-	-	Input Sukses	Input Sukses
				√ ⁽²⁾	√	√	-	-	-	Error: Kode materi telah digunakan	Error: Kode materi telah digunakan
				-	√	√	√	-	-	Error: Input tidak lengkap	Error: Input tidak lengkap
				-	-	√	√	√	-	Error: Input tidak lengkap	Error: Input tidak lengkap
				√ ⁽¹⁾	√	√	√	-	-	Input Sukses	Input Sukses
				√ ⁽²⁾	√	√	√	-	-	Error: Kode materi telah digunakan	Error: Kode materi telah digunakan
				-	√	√	√	√	-	Error: Input tidak lengkap	Error: Input tidak lengkap
				√ ⁽¹⁾	√	√	√	√	-	Input Sukses	Input Sukses
				√ ⁽²⁾	√	√	√	√	-	Error: Kode materi telah digunakan	Error: Kode materi telah digunakan
5	4	Menampilkan daftar kurikulum semester ke-5	Menampilkan daftar kurikulum semester ke-5								
6	3	Menampilkan daftar kurikulum semester ke-6	Menampilkan daftar kurikulum semester ke-6								
7	2	Menampilkan daftar kurikulum semester ke-7	Menampilkan daftar kurikulum semester ke-7								
8	1	Menampilkan daftar kurikulum semester ke-8	Menampilkan daftar kurikulum semester ke-8								
9	(-)	Tidak ada tampilan	Tidak ada tampilan								
10	(-)	Tidak ada tampilan	Tidak ada tampilan								
11	(-)	Tidak ada tampilan	Tidak ada tampilan								

Keterangan:

* : nilai awal

(-) : tidak melewati *looping* karena $\$i > 8$

4.5 Pengujian *Black Box*

Metode pengujian *black box* merupakan metode pengujian dengan pendekatan yang mengasumsikan sebuah sistem perangkat lunak atau program sebagai sebuah kotak hitam (*black box*). Pendekatan ini hanya mengevaluasi program dari output atau hasil akhir yang dikeluarkan oleh program tersebut.

Pengujian *black box* pada makalah tugas akhir ini hanya dilakukan pada proses penginputan kurikulum, dengan hasil pengujian yang dapat dilihat pada tabel 4.3.

Tabel 4.3 Hasil pengujian *black box* pada proses penginputan kurikulum

Input					Hasil yang diharapkan	Output sistem
Kode materi	Nama materi	Semester	Ringkasan	Referensi		
-	-	-	-	-	Error: Input tidak lengkap	Error: Input tidak lengkap
√	-	-	-	-	Error: Input tidak lengkap	Error: Input tidak lengkap
-	√	-	-	-	Error: Input tidak lengkap	Error: Input tidak lengkap
-	-	√	-	-	Error: Input tidak lengkap	Error: Input tidak lengkap
-	-	-	√	-	Error: Input tidak lengkap	Error: Input tidak lengkap
-	-	-	-	√	Error: Input	Error: Input

Keterangan:

√ : data diisi

- : data kosong

(1) : kode mata kuliah belum digunakan

(2) : kode mata kuliah telah digunakan

V. PENUTUP

5.1 Kesimpulan

1. Sistem *e-learning* harus didukung oleh sumber daya yang berkualitas agar proses belajar mengajar yang terjadi di dalamnya dapat efektif.
2. Sistem *e-learning* yang dibangun dalam tugas akhir ini hanya digunakan sebagai penunjang mentoring agama Islam dan tidak diterapkan sebagai pengganti mentoring agama Islam konvensional.
3. Berdasarkan hasil pengujian dengan metode *black box*, sistem *e-learning* yang dibangun dalam tugas akhir ini telah sesuai dengan yang diharapkan pada awal perancangan.
4. Berdasarkan hasil pengujian dengan metode *white box*, kode program dalam sistem *e-learning* telah cukup efektif baik dari segi alur program, pernyataan kondisional, maupun *looping* yang terdapat di dalamnya.
5. Perancangan sistem yang matang akan menghasilkan program yang baik dan sesuai dengan yang diinginkan.
6. Penggunaan *use case* yang baik akan membantu *programmer* untuk membuat program yang sesuai dengan keinginan klien.

5.2 Saran

1. Perlu ada pengembangan fasilitas penunjang *e-learning*, seperti multimedia dengan *webcam*, *microphone* dan lainnya.
2. Sekuritas sistem dapat lebih ditingkatkan agar data-data penting yang ada dalam sistem terjaga. Perkembangan teknologi informasi memungkinkan suatu sistem dapat dijebol oleh orang tidak bertanggungjawab.

3. Dalam penulisan kode program harus efisien dan efektif, sehingga program yang dijalankan berfungsi dengan baik dan tidak menampilkan banyak baris kode program.
4. Sebelum menuliskan kode program diharapkan perancangan dari keseluruhan sistem telah baik, sehingga tidak ada perubahan yang mendasar ketika sudah menuliskan kode program.

**Muhammad Guruh Wicaksana
(L2F 001 621)**

Lahir sebagai anak kedua dari 3 bersaudara pada Tanggal 12 Desember 1982 di Ciamis. Sedang melanjutkan studi di S1 Teknik Elektro UNDIP angkatan tahun 2001. Mengambil konsentrasi Informatika dan Komputer.

DAFTAR PUSTAKA

- [1] Azis M.Kom, Ir. M. Farid, *Object Oriented Programming dengan PHP 5*, Penerbit PT. Elex Media Komputindo Jakarta 2005
- [2] Cosentino, Christopher, *Advance PHP for Web Professionals*, Prentice Hall PTR 2002
- [3] Gilfillan, Ian, *Mastering MySQL 4*, Sybex 2003
- [4] Greenspan, Jay, Brad Bulger, *MySQL/PHP Database Applications*, IDG Books Worldwide.Inc
- [5] Hakim , Lukmanul dan Uus Musalini, *Cara Mudah Memadukan Web Design Dan Web Programming*, Penerbit PT. Elex Media Komputindo Jakarta 2004
- [6] Jarger, Randy J, George Reese, Tim King, Hugh E. Williams, *Managing and Using MySql, 2nd Edition*, O'reilly 2002
- [7] Jasmadi, *Koleksi Template Web dan Teknik Pembuatannya*, Penerbit ANDI Yogyakarta 2004
- [8] Lerdorf, Rasmus, *PHP Pocket Reference, 2nd Edition*, O'Reilly 2002
- [9] Meloni, Julie. C, *Sams Teach Yourself PHP, MySQL and Apache in 24 Hours*, Sams Publishing 2002
- [10] Naidu Ph.D, Som, *E-Learning A Guidebook of Principle, Procedure and Practices*, Commonwealth Educational Media Centre Of Asia (CEMCA) 2003
- [11] Pressman Ph.D, Roger S, *Rekayasa Perangkat Lunak Pendekatan Praktisi Buku 1 dan 2*, Penerbit ANDI Yogyakarta 2003
- [12] Siahaan, Sudirman, *E-Learning (Pembelajaran Elektronik) Sebagai Salah Satu Alternatif Kegiatan Pembelajaran*, Pusat Data dan Informasi Pendidikan, Balitbang - Depdiknas 2001
- [13] Siswoutomo, Wiwit, *PHP Undercover Mengungkap Rahasia Pemrograman PHP*, Penerbit PT. Elex Media Komputindo Jakarta 2005
- [14] Syafii, M, *Membangun Aplikasi berbasis PHP dan MySql*, Penerbit ANDI Yogyakarta 2004
- [15] Wiilyanto,S.T, *Perancangan Aplikasi E-Education Berbasis Web Dengan Menggunakan Bahasa Pemrograman PHP*, Tugas Akhir, 2005

Mengetahui/mengesahkan,

Dosen Pembimbing I

Dosen Pembimbing II

Maman Somantri, ST, MT
NIP. 132 231 133

Agung B.P, ST, MIT
NIP. 132 137 932