

PERANCANGAN APLIKASI M-LEARNING BERBASIS JAVA

Bambang Riyanto, briyanto@lisk-svr1.ee.itb.ac.id
Muh. Tamimuddin H., muh_tamim@yahoo.com
Sri Widayati, widwid79@yahoo.com

Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung

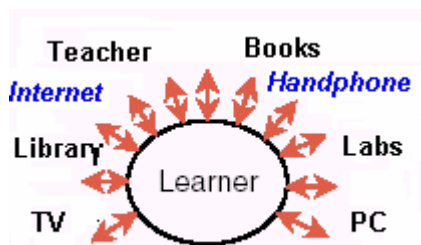
ABSTRAK

Teknologi telekomunikasi dan teknologi internet menjadi gelombang kecenderungan baru di seluruh dunia. Perkembangan ini memungkinkan terobosan baru dalam belajar secara mobile menggunakan perangkat IT genggam atau disebut mobile learning (m-learning). M-learning memiliki beberapa kelebihan di antaranya adalah kemampuan belajar “kapan-pun di mana-pun”. Namun, di sisi lain, perangkat pembelajaran m-learning memiliki keterbatasan sumber daya dan keragaman platform sehingga diperlukan rancangan yang mampu menjamin kompatibilitas dan interoperabilitas.

Kata kunci: m-learning, e-learning, OOP, Java, XML, XML-RPC

1. PENDAHULUAN

Perkembangan teknologi telah menciptakan pengembangan terobosan-terobosan dalam pembelajaran. Di tengah perkembangan ini learner (pembelajar) bersinggungan dengan perangkat-perangkat teknologi komunikasi bergerak dan teknologi internet telah menjadi gelombang kecenderungan baru yang memungkinkan pembelajaran secara mobile atau lebih dikenal sebagai mobile learning (m-learning). Kombinasi teknologi telekomunikasi dan internet memungkinkan pengembangan sistem m-learning yang pada sisi klien memanfaatkan divais bergerak, berinteraksi dengan sisi server, yaitu web server.



GAMBAR 1.1. INTERAKSI LEARNER DENGAN LINGKUNGAN

Meskipun saat ini m-learning masih berada pada tahap awal pengembangan dan para peneliti masih mengeksplorasi setiap aspeknya, m-learning diperkirakan akan menjadi cukup pesat dan viabel

dalam jangka waktu dekat[2]. Hal ini didukung oleh beberapa faktor berikut.

- Divais makin banyak, murah dan canggih serta kenyataan bahwa pengguna divais bergerak lebih banyak dari pengguna komputer.
- Perkembangan teknologi wireless/seluler (2G, 2.5G, 3G).
- Tuntutan kebutuhan konsumen.

Tidak seperti e-learning tradisional, sumber daya (computing resources) pada lingkungan m-learning sangat terbatas. Divais bergerak yang digunakan sebagai media belajar memiliki beberapa keterbatasan, seperti catu daya, kapasitas penyimpanan, pemroses, layar tampilan dan sarana masukan/keluaran. Di samping itu, divais bergerak memiliki platform yang beragam, begitu pula dengan platform server yang dipakai sebagai sumber daya pembelajaran. Kenyataan ini menyebabkan sistem m-learning harus dikembangkan secara khusus dan dioptimasi sedemikian rupa untuk dapat kompatibel pada divais yang beragam dan sumber daya yang terbatas serta memiliki interoperabilitas yang tinggi.

Java yang merupakan open standard yang portable telah memberikan dukungan bagi pengembangan aplikasi yang beragam, dari aplikasi enterprise berbasis web menggunakan J2EE sampai aplikasi yang berjalan pada perangkat bergerak dengan J2ME dan dapat berjalan pada banyak platform mesin, sistem operasi dan divais. Implementasi m-learning menggunakan

Java dapat menjamin kompatibilitas yang tinggi mengingat aplikasi Java dapat ditanam pada divais dan mesin yang beragam. Untuk mengembangkan aplikasi Java ini diperlukan metodologi perancangan berorientasi obyek (OOD). Sedangkan untuk menjamin adanya interoperabilitas digunakan teknologi XML yang telah menjadi sebuah standar pertukaran data yang sudah cukup mapan (*mature*) dan telah diterima secara luas. Teknologi XML yang diimplementasikan pada sistem m-learning menawarkan sistem yang lebih terbuka, bebas platform dan interoperabel. Untuk mendukung semua ini diperlukan arsitektur pembelajaran yang netral dan dapat diadopsi secara fleksibel.

Pada penelitian ini proses pengembangan aplikasi menggunakan RUP (*Rational Unified Process*) sebagai metodologi dan UML (*Unified Modelling Language*) sebagai bahasa untuk memodelkan sistem. Protokol yang dipakai untuk berkomunikasi digunakan XML-RPC (*XML Remote Procedure Call*). Sedangkan acuan arsitektur pembelajaran menggunakan LTSA (*Learning Technology System Architecture*).

2. M-LEARNING

Istilah *mobile learning (m-learning)* mengacu kepada penggunaan perangkat IT genggam dan bergerak, seperti PDA, telepon genggam, laptop dan tablet PC, dalam pengajaran dan pembelajaran [1]. M-learning adalah pembelajaran yang unik karena pembelajar dapat mengakses materi pembelajaran, arahan dan aplikasi yang berkaitan dengan *course* kapan-pun dan dimana-pun. Hal ini akan meningkatkan perhatian pada materi pembelajaran, membuat pembelajaran menjadi pervasif, dan dapat mendorong motivasi pembelajar kepada pembelajaran sepanjang hayat (*lifelong learning*). Selain itu, dibandingkan pembelajaran konvensional, m-learning memungkinkan adanya lebih banyak kesempatan untuk kolaborasi secara *ad hoc* dan berinteraksi secara informal di antara pembelajar [2].

Dalam pembelajaran e-learning, independensi waktu dan tempat menjadi faktor penting yang sering ditekankan. Namun, dalam e-learning tradisional kebutuhan minimum tetap sebuah PC yang dengan demikian memiliki konsekuensi bahwa independensi waktu dan tempat tidak sepenuhnya terpenuhi [2]. Pembelajaran m-learning menjanjikan independensi waktu dan tempat dalam arti sesungguhnya.

Meski memiliki beberapa kelebihan, m-learning tidak akan sepenuhnya menggantikan e-learning tradisional. Dengan keterbatasan *computing resources*, m-learning tidak dapat menyediakan dan/atau mengakses sumber daya pembelajaran yang sama dengan sumber daya

pada e-learning. M-learning akan sangat mungkin untuk digunakan sebagai suplemen bagi e-learning maupun bagi lingkungan pembelajaran tradisional di mana *computer aided learning* tidak tersedia [2]. Sistem yang paling optimal adalah menggabungkan m-learning dengan e-learning, di mana ada alternatif proses pembelajaran dilakukan dengan perangkat komputer dan/atau divais bergerak [2] atau dikombinasikan dengan sistem pembelajaran tradisional.

2.1. Aspek Rancangan M-Learning

Karakteristik perangkat maupun pengguna m-learning yang khusus dan unik memerlukan disain yang juga khusus. Beberapa aspek yang menjadi perhatian dalam merancang aplikasi m-learning adalah sebagai berikut.

- 1. Keterbatasan Hardware.** Dengan keterbatasan *computing resources* hardware ini hal-hal yang perlu diperhatikan adalah perlunya pemilihan penggunaan pustaka yang tepat, pengurangan *footprint* aplikasi, serta pemanfaatan portal yang mendelegasikan pekerjaan-pekerjaan kompleks dan memerlukan sumber daya besar [4].
- 2. Keterbatasan Jaringan.** Jaringan seluler memiliki kekurangan di antaranya adalah sangat lambat, tak dapat diandalkan dan tidak aman. Keadaan ini menyebabkan pengembangan aplikasi m-learning harus memanfaatkan sumber daya jaringan semaksimal mungkin. Aplikasi harus mampu memberi dukungan operasi secara *offline* sehingga tidak harus terhubung secara terus menerus dengan server serta mekanisme download yang dapat diandalkan. Jika memerlukan koneksi secara, sedapat mungkin digunakan cara yang efisien. Sedangkan ketidak-amanan data dapat di atasi dengan implementasi enkripsi [4].
- 3. Divais yang Pervasif.** Pengelolaan devais kecil yang beragam dan dapat selalu dibawa ke mana-mana oleh pengguna dan dengan mudah berpindah tangan merupakan persoalan yang lebih rumit, baik persoalan sosial maupun persoalan teknis. Kenyataan ini memerlukan adanya mekanisme proteksi *on-device* untuk melindungi data sensitif. Selain itu diperlukan adanya optimasi bagi aplikasi untuk dijalankan pada platform berbeda (*device-independent*) [4].
- 4. Skema Integrasi.** Banyak aplikasi nirkabel bergerak yang membutuhkan integrasi dengan banyak sistem *back-end* atau *middleware* berbeda. Saat ini terdapat beberapa teknologi yang dapat digunakan, diantaranya adalah Protokol biner *proprietary*, Framework RPC, *messaging* serta XML *web services*. Masing-masing teknologi memiliki kelebihan dan kekurangan masing-

masing. Semakin longgar kopling integrasinya (*loosly coupled*) akan semakin besar pula *footprint* dan ukuran aplikasi serta data yang ditransfer [4].

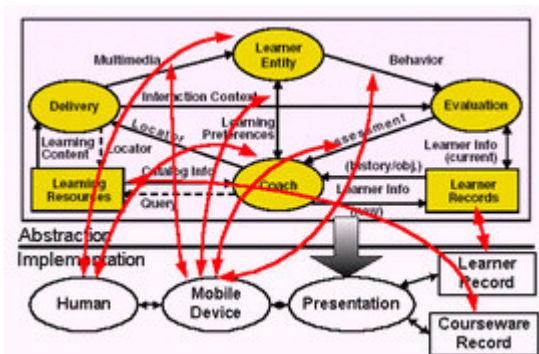
5. Kenyamanan Pengguna. Beberapa hal yang perlu untuk diperhatikan adalah perlunya antarmuka yang kaya, tampilan yang tidak terlalu padat, pemanfaatan *thread* untuk proses yang lama, penyimpanan preferensi pengguna dan penyediaan *deployment descriptor* untuk kemudahan instalasi [4][2].

3. ANALISIS SISTEM

3.1. Sistem M-Learning Yang Dikembangkan

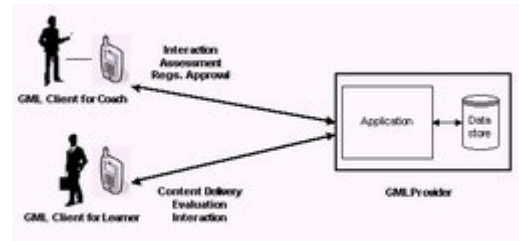
Sistem yang dikembangkan dalam penelitian ini bernama **Ganesa Mobile Learning (GML)**. Sistem ini terbagi menjadi dua buah sub-sistem, yaitu **GML Provider** dan **GML Client**. GML Provider merupakan sebuah sub-sistem GML yang dijalankan pada web server yang menyediakan *service* m-learning yang dapat diakses dan dijalankan oleh sub-sistem lain, yaitu GML Client, yang berupa *smartphone*, melalui HTTP. Rancangan kedua sub-sistem ini menggunakan acuan *Object Oriented Design (OOD)* sehingga hasil rancangan dapat diimplementasikan pada platform Java.

Arsitektur pembelajaran GML mengacu kepada arsitektur *Learning Technology System Architecture (LTSA)* yang usulkan oleh IEEE 1484 *Learning Technology Systems Committee (LTSC)*. Standar ini meliputi wilayah yang luas dari sistem, umumnya dikenal sebagai teknologi pembelajaran, teknologi pendidikan dan pelatihan, pelatihan berbasis komputer (CBT), *computer assisted instruction*, *intelligent tutoring*, metadata, dan sebagainya. Standar ini netral secara pedagogis, netral terhadap konten, netral secara kultural dan netral platform [5]. Arsitektur LTSA ini dipetakan ke dalam lingkungan pembelajaran m-learning (Gambar 3.1).



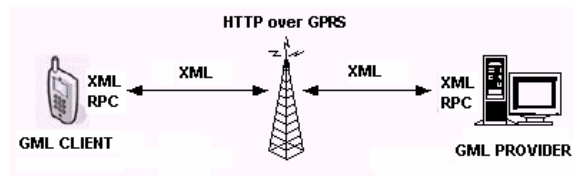
GAMBAR 3.1. PEMETAAN LTSA KE ARSITEKTUR GML

Pada rancangan GML ini beberapa karakteristik yang membedakan dengan sistem e-learning adalah pada penggunaan *mobile device* yang dipakai oleh *learner*, seperti untuk mengakses *learning content* atau dalam berinteraksi antar *learner* maupun dengan *coach*.



GAMBAR 3.2. ARSITEKTUR PEMBELAJARAN GML

Secara fisik, arsitektur GML dibagi menjadi dua bagian, yaitu *GML Provider* dan *GML Client*. Pertukaran pesan antara *GML Provider* dan *GML Client* protokol berbasis XML, yaitu XML-RPC sederhana dan telah diimplementasikan pada banyak bahasa [6]. Penggunaan protokol ini dimaksudkan agar pengembangan dua subsistem ini dapat bersifat *loosly couple*. Implementasi protokol XML-RPC sendiri cukup ringan sehingga aplikasi yang dihasilkan dapat berukuran relatif kecil, data yang ditransfer juga relatif kecil serta kecepatan lebih tinggi dibanding protokol sejenis [7]. Sedangkan protokol *transport* menggunakan protokol HTTP over GPRS yang sudah cukup mapan (*well established*) dan diterima luas.



GAMBAR 3.3. ARSITEKTUR FISIK GML

3.2. Skenario

Pada sistem ini terdapat tiga pengguna, yaitu *learner* (pelajar), *coach* (instruktur) dan administrator sistem. Skenario belajar secara umum pada sistem GML ini adalah sebagai berikut.

1. Learner melakukan registrasi ke sistem untuk mendapatkan *username* dan *password*.
2. Dengan *username* dan *password* yang telah didapatkan, learner melakukan login. Jika login diterima ia akan mendapatkan sebuah *session ID* sebagai identitas unik dari setiap sesi belajar dan akan selalu digunakan sebagai identitas learner untuk masuk ke sistem. *Session ID* ini akan tetap aktif sampai learner melakukan logout.

3. Setelah melakukan login ke sistem, learner melihat daftar course yang ditawarkan sistem. Learner memilih salah satu course yang ditawarkan. Jika permintaan disetujui atau course termasuk kategori bebas diikuti (tanpa perlu *approval* dari sistem atau pre-test) maka course tersebut akan dimasukkan ke daftar course yang diikuti oleh learner.
4. Learner melihat daftar course yang diikuti. Ia dapat memilih salah satu course, dan men-download materi course dari server, menyimpannya di divais dan memulai pembelajaran.
5. Learner dapat mengikuti ujian akhir (*examination*) untuk menguji tingkat pemahamannya. Jika learner dinyatakan lulus, data kelulusan/*assesment* dimasukkan ke data portofolio learner.
6. Selama mengikuti course, learner dapat melakukan diskusi dengan sesama learner yang mengikuti course yang sama atau dengan coach dengan mengirimkan teks pertanyaan atau tanggapan ke forum.
7. Learner yang menyelesaikan sebuah sesi pembelajaran dapat melakukan logout. Proses ini akan me-reset session ID yang didaftarkan di sistem.

Dari beberapa proses tersebut, aplikasi hanya memerlukan koneksi ke server pada saat dibutuhkan, yaitu ketika login, melihat daftar course dan konten di server, mendownload konten, mengirim jawaban soal, dan saat mengirim pesan.

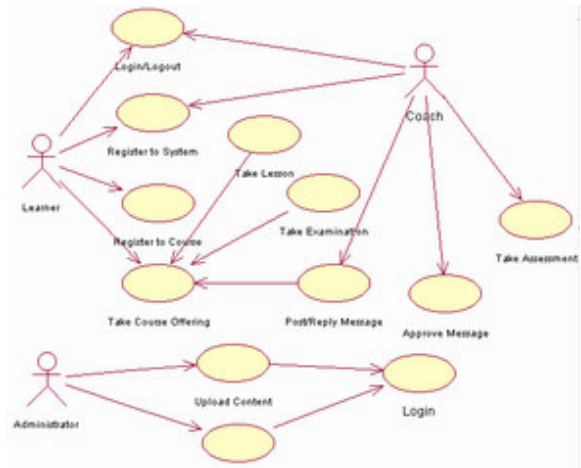
3.3. Fungsi Sistem

Mendefinisikan fungsi dari sistem merupakan langkah awal dalam analisis sistem. Fungsi-fungsi yang dapat diidentifikasi dari GML adalah sistem mampu untuk:

- menyediakan fungsi registrasi sistem,
- melayani fungsi registrasi course,
- menyimpan data course dan data learner yang mengikuti course tersebut,
- mengirimkan konten pembelajaran yang tepat,
- menyediakan ujian dan mekanisme untuk evaluasi,
- mengolah hasil evaluasi,
- menyediakan fasilitas interaksi antar sesama learner maupun dengan coach.

3.4. Use-Case Diagram

Use-case diagram menggambarkan aktor eksternal dan hubungannya dengan *use-case*. *Use-case* mendeskripsikan fungsi yang disediakan sistem.



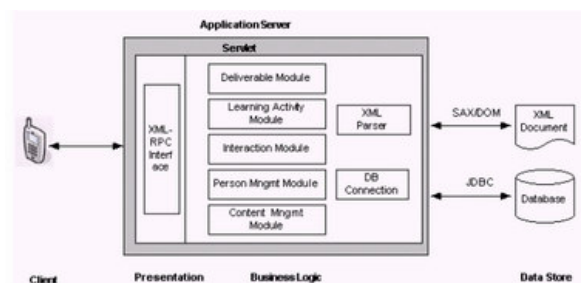
GAMBAR 2.4. USE CASE DIAGRAM

4. DISAIN SISTEM

4.1 Disain Sub-Sistem GML Provider

4.1.1 Disain Arsitektur GML Provider

Disain arsitektur GML Provider mengacu pada *three-tier architecture*. Dalam arsitektur ini bagian-bagian sistem dipisahkan antara *presentation*, *business logic* dan *data store* (Gambar 3.1.). Lapisan *presentation* pada sistem ini adalah XML-RPC interface yang merupakan kumpulan prosedur yang di-*publish* dan dapat dipanggil oleh aplikasi lain secara *remote* melalui protokol HTTP. Servis GML Provider tidak dimaksudkan untuk dikonsumsi pengguna secara langsung.



GAMBAR 4.1. ARSITEKTUR SUB-SISTEM GML PROVIDER

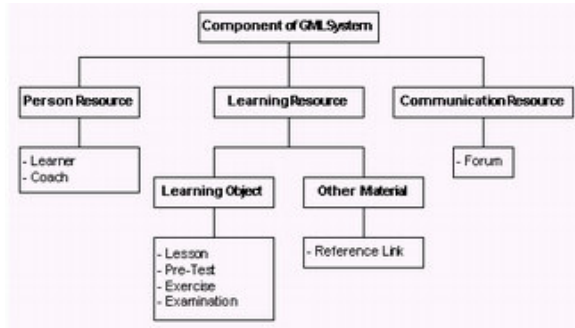
Lapisan *business logic tier* merupakan bagian yang menangani proses-proses inti dari sistem. Lapisan ini terdiri dari beberapa modul, yaitu *deliverable module*, *interaction module*, *learning activity module*, *person management module* dan *content management module*.

Lapisan *data store* adalah lapisan penyimpanan data sistem. Pada sistem GML Provider data yang digunakan sebagai sumber data layanan disimpan dalam database serta dokumen XML. Untuk mengakses

database digunakan JDBC, sedangkan untuk mengakses dokumen XML digunakan SAX dan DOM.

4.1.2 Disain Komponen

Secara umum, komponen dari GML Provider digambarkan oleh Gambar 4.2.

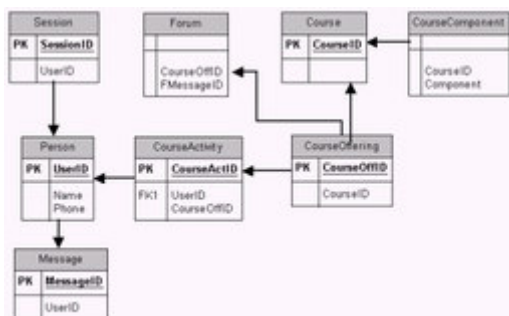


GAMBAR 4.2. KOMPONEN GML PROVIDER

Sistem GML menggunakan *learning object* berupa file XML yang di-upload dan dikelola di server. Penambahan atau perubahan materi dilakukan dengan membuat atau mengedit file XML ini. Materi pembelajaran dapat dimasukkan dalam file XML atau diletakkan di lokasi lain, namun tetap dirujuk dari file ini, artinya jika *learning object* yang digunakan berada di file lain, maka dalam file XML harus ditunjukkan rujukan/link ke file tersebut.

4.1.3. Disain Database GML Provider

Informasi selain *learning object* disimpan dalam sebuah database relasional. Database GML Provider diilustrasikan oleh Gambar 4.3.



GAMBAR 4.3. DISAIN DATABASE GML PROVIDER

4.1.4. Disain XML-RPC Interface

XML-RPC Interface merupakan *service endpoint interface* yang mendeklarasikan metoda/prosedur yang di-publish dan dapat dipanggil oleh *remote client* dan merupakan pemetaan dari *use case diagram*. Aktor

yang dapat memanfaatkan servis ini hanya learner dan coach. Sedangkan administrator memiliki mekanisme sendiri, yaitu melalui akses web biasa (tanpa XML-RPC). *XML-RPC Interface* yang didefinisikan pada sub-sistem GML Provider ditunjukkan oleh Tabel 4.1.

TABEL 4.1. XML-RPC INTERFACE

Use Case	Aktor	XML-RPC Interface
Login/logout	Learner Coach	login() logout()
Register to System	Learner,Coach	registerSystem()
Register to Course	Learner	listCourse() registerCourse()
Take CourseOffering	Learner	selectCourseOff() listCourseOff()
Take Lesson	Learner	selectLesson()
Take Pre-Test	Learner	selectPreTestSheet() makePreTestSolution()
Take Examination	Learner	selectEmSheet() makeEmSolution()
Take Reference Link	Learner	displayRefLink()
Read/Send Forum Message	Learner Coach	readMessage() sendMessage() approveMessage()
Approve Forum Message	Coach	listPendingMessage() approveMessage()
Approve Registrar	Coach	listPendingRegistrar() approveRegistrar()
Take Assesment	Coach	assessResult() displayLnrAssessment()
Take Private Message	Learner	readPrivatMessage ()
Make Announcement	Coach	sendAnnouncement()

4.1.5. Model Konseptual GML Provider

Kandidat konsep/kelas dari sistem dapat diidentifikasi dari rancangan *use case* dan rancangan komponen yang telah dibuat. Daftar konsep ditunjukkan Gambar 4.4.

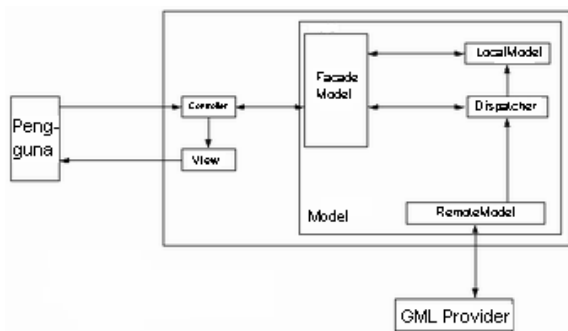


GAMBAR 4.4. KONSEP DALAM GML PROVIDER

Model konseptual ini merupakan landasan bagi pengembangan selanjutnya dengan merancang *interaction diagram*, *class diagram*, *component diagram* dan *deployment diagram* serta selanjutnya untuk diimplementasikan di lingkungan pemrograman *Java Enterprise Edition (J2EE)* dan pustaka *Apache XML-RPC* (<http://ws.apache.org/xmlrpc/>).

4.2 Arsitektur Sub-Sistem GML Client

Sub-sistem GML Client difokuskan untuk menjembatani keseluruhan sistem dengan pengguna (*end user*). Beberapa kompleksitas data dan pemrosesan dapat dikurangi secara signifikan dengan memanfaatkan servis XML-RPC dari GML Provider. Namun demikian, sub-sistem GML Client masih memerlukan manajemen pengelolaan data pada divais, utamanya untuk mendukung aplikasi yang dapat dijalankan pada mode offline dan terkoneksi dengan server hanya pada saat dibutuhkan (*occasionally connected*). Arsitektur GML Client mengacu kepada arsitektur MVC (*Model View Controller*) dan ditunjukkan oleh Gambar 4.5.



GAMBAR 4.5. ARSITEKTUR SUB-SISTEM GML CLIENT

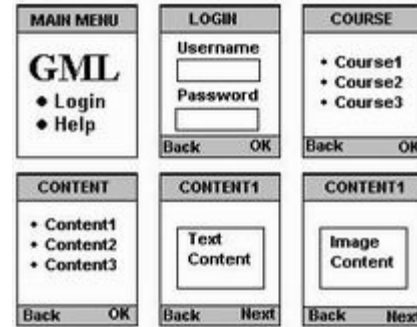
Arsitektur model sub-sistem GML Client terdiri dari:

- *FacadeModel*, merupakan komponen menangani semua aksi dari komponen *model* ke *controller*.
- *LocalModel*, merupakan komponen yang bertugas untuk menangani aksi yang mengakses media penyimpanan lokal pada device.
- *Dispatcher*, merupakan komponen yang bertugas untuk menangani pemilahan data.
- *RemoteModel*, merupakan komponen yang bertugas untuk menangani aksi yang memerlukan akses ke remote server. *RemoteModel* merupakan XML-RPC Interface.

Dari komponen-komponen ini kemudian dilakukan pengembangan komponen yang lebih detail dan penambahan komponen lain yang diperlukan, seperti komponen antarmuka.

4.2.1. Disain Antarmuka

Sebagai media yang berinteraksi langsung dengan pengguna, GML Client harus memiliki antarmuka yang tepat, ramah dan efektif. Gambar 4.6. menunjukkan sebagian antarmuka GML Client.



GAMBAR 4.6. ANTARMUKA GML CLIENT

4.2.2. Model Konseptual GML Client

Konsep dari sub-sistem GML Client yang diidentifikasi ditunjukkan oleh Gambar 4.7.



GAMBAR 4.7. KONSEP DALAM GML CLIENT

Model konseptual ini menjadi dasar pengembangan lebih lanjut. GML Client menggunakan perangkat divais bergerak dan diimplementasikan pada *Java Micro Edition* (J2ME) dengan tambahan pustaka *kxmlrpc* (<http://kxmlrpc.enhydra.org>).

5. KESIMPULAN

Dari uraian di atas dapat diambil beberapa kesimpulan berikut.

1. M-Learning merupakan pembelajaran yang cukup prospektif dan *viable* untuk diimplementasikan.
2. Kendala pengembangan aplikasi m-learning adalah keterbatasan sumber daya dan keragaman platform sehingga perlu rancangan yang mampu mengatasi kendala ini sehingga dapat dikembangkan aplikasi yang memiliki kompatibilitas dan interoperabilitas tinggi.
3. Rancangan m-learning yang menggunakan metodologi RUP dan berorientasi obyek dapat diimplementasikan ke platform Java sehingga dapat di-*deploy* dan kompatibel dengan divais, mesin dan sistem operasi yang beragam.

4. Protokol XML-RPC yang dipakai sebagai protokol komunikasi antara klien dan server memungkinkan pengembangan yang *loosly coupled* dan memiliki interoperabilitas tinggi.

Informasi lebih lanjut mengenai pengembangan GML dapat ditemukan di website <http://www.gmobile.co.nr>

6. REFERENSI

[1] Wood, Karen, *Introduction to Mobile Learning (M Learning)*, Ferl, Becta (British Educational Communications and Technology Agency), 2003. <http://ferl.becta.org.uk/display.cfm?page=65&catid=192&resid=5194>. Diakses tanggal 19 Nopember 2005.

[2] Holzinger, Andreas, Alexander Nischelwitzer, dan Matthias Meisenberger, *Mobile Phones as a Challenge for m-Learning: Experiences with the Mobile Learning Engine (MLE) using Mobile Interactive Learning Objects (MILOs)*, http://dmt.fh-joanneum.at/kd3/objects/application_pdf/percom_2004_%20hawaii_PerIL_ah_nis.pdf. Diakses tanggal 23 Februari 2006.

[3] Virvon, Maria dan Eythimios Alepis, *Mobile versus desktop facilities for an e-learning system: users' perspective*, IEEE International Conference on Systems, Man and Cybernetica, 2004.

[4] Juntao Yuan, Michael, *Enterprise J2ME : Developing Mobile Java Applications*, Prentice Hall, 2004.

[5] _____, *IEEE P1484.1/D9, 2001-11-30 Draft Standard for Learning Technology*, Learning Technology System Architecture (LTSA), <http://edutools.com/ltsa>

[6] _____, *XML-RPC Spesification*, Userland Software, <http://www.xmlrpc.com>. Diakses tanggal 19 November 2005.

[7] _____, *Optimizing the Client/Server Communication for Mobile Applications*, Part 3, 2003, <http://forum.nokia.com>. Diakses tanggal 23 Februari 2006.

