

## **Pemanfaatan Basis Data *Mobile* dan Teknik Sinkronisasi dalam Model Pembelajaran *Mobile***

### ***Utilization mobile Database and Synchronization Techniques on Mobile Learning Model***

**Annisa Dwi Anggraini, Azhari**

Magister Manajemen Informasi, FMIPA UGM, Yogyakarta  
e-mail: nissa.dwi@gmail.com

#### **Abstrak**

Learning Management System (LMS) merupakan media pembelajaran yang memiliki keterbatasan dalam mengaksesnya dan tergantung pada ketersediaan koneksi internet dan perangkat desktop yang. Untuk itu, diperlukan upaya membangun aplikasi yang khusus untuk perangkat mobile sehingga LMS dapat diakses melalui perangkat mobile dan tidak selalu harus terhubung ke internet.

Pengembangan aplikasi perangkat mobile yang dapat mengakses LMS pada sisi server dan mereplikasi data dilakukan melalui middleware untuk menyinkronkan data berdasarkan teknik sinkronisasi SyncML. Data yang telah disinkronisasikan disimpan dalam database lokal SQLite yang telah terintegrasi dalam aplikasi. Proses sinkronisasi dan ketersediaan database mobile memungkinkan aplikasi dapat diakses secara online atau offline tanpa harus koneksi ke internet.

Hasil pengujian aplikasi pada berbagai perangkat mobile memberikan kesimpulan di mana spesifikasi yang lebih tinggi perangkat mobile yang digunakan, semakin sedikit waktu untuk melakukan sinkronisasi data. Selain itu, kebutuhan untuk sinkronisasi data juga tergantung pada ukuran data di mana semakin besar ukuran data yang akan disinkronisasi, maka semakin lama waktu untuk menyelesaikan proses sinkronisasi data.

**Kata kunci** : mobile application, synchronization, SyncML, local database.

#### **Abstract**

Learning Management System (LMS) as a medium of learning has a limitation in which to access it depends on the availability of internet connection and desktop device. For that reason, it is necessary to build an application that is specific to mobile devices so that the LMS can be accessed through mobile devices and do not always have to be connected to the internet.

Application development of mobile devices can access the LMS on the server side and replicate data through middleware to synchronize data based on SyncML synchronization techniques. The data **have** been synchronized stored in SQLite local database that has been integrated into the application. The synchronization process and the availability of mobile database enable the application can be accessed online or offline without having a connection to the internet.

Results of testing applications on a variety of mobile devices give conclusion where the higher specification mobile devices are used, the less time to perform data synchronization. Also, the need for data synchronization also depends on the size of the data where the larger the size of data to be synchronized, then the longer the time for completing the data synchronization process.

**Keywords**: mobile application, synchronization, SyncML, local database.

## 1. Pendahuluan

Learning Management System(LMS) merupakan salah satu penerapan teknologi internet dalam pembelajaran. Almrashdeh, dkk 2011 menyatakan bahwa LMS merupakan salah satu teknologi internet yang dapat digunakan sebagai perangkat pembelajaran baik untuk pengajar maupun pelajar dimana saat ini LMS telah banyak dimanfaatkan pada lingkungan universitas. Di sisi pengajar yaitu dosen, LMS memungkinkan pengguna untuk berbagi materi, mengatur kelas, memberikan tugas dan memberikan evaluasi. Di sisi lain, LMS memberikan kemudahan bagi pelajar atau mahasiswa untuk mendownload materi, berdiskusi, mengerjakan dan mengupload tugas dan mengetahui performa belajar mereka.

LMS untuk keperluan pembelajaran tidak hanya dapat diakses melalui desktop dan komputer, tetapi juga melalui perangkat bergerak. Permasalahan muncul ketika pengguna LMS tidak mendapatkan koneksi internet dan tidak tersedia perangkat desktop untuk mengakses LMS. Oleh karena itu, pemanfaatan LMS bergantung pada koneksi internet dan perangkat yang digunakan (Rahman, dkk, 2010).

Keterbatasan tersebut memberikan kemungkinan bagi pengguna untuk mengakses LMS melalui perangkat lain seperti perangkat bergerak. Namun dengan keterbatasan yang dimiliki perangkat bergerak seperti keterbatasan daya baterai maka penggunaan LMS tidak lebih leluasa daripada menggunakan perangkat desktop. Permasalahan pada perangkat bergerak tersebut memunculkan peluang untuk membangun aplikasi berbasis *mobile* yang diperuntukkan dalam mengakses LMS. Aplikasi *mobile* untuk kepentingan pembelajaran biasanya disebut dengan *mobile learning*. Dari sisi portabilitas dan fleksibilitas, *m-learning* lebih baik daripada *e-learning* karena *m-learning* diakses melalui perangkat bergerak sehingga memungkinkan seseorang untuk belajar dimana saja dan kapan saja (Sarrab, dkk, 2012)

Permasalahan tersebut memunculkan ide untuk mengembangkan aplikasi *mobile learning* yang memiliki sifat perpaduan antara *online* dan *offline mobile application*. Pengembangan aplikasi *mobile learning* tersebut dimaksudkan agar pengguna tidak perlu harus bergantung pada perangkat desktop dan koneksi internet dalam mengakses LMS. Penelitian ini mencoba untuk membangun aplikasi model pembelajaran *mobile* secara *online* dan *offline*. Aplikasi yang dibangun menggunakan *mobile database* dan terhubung dengan basis data *server* saat perangkat mengakses internet untuk mengambil data atau melakukan sinkronisasi data dan menyimpannya di *mobile database* agar dapat diakses secara *offline*.

## 2. Studi Pustaka

Beberapa penelitian mengenai pengembangan aplikasi untuk perangkat bergerak dan layanan seperti sinkronisasi dan penggunaan *mobile database* telah beberapa kali dilakukan (Alqahtan dan Goodwin, 2012; Boyinbodedan Akinyede, 2008). Penelitian terkait dengan sinkronisasi data pada perangkat bergerak dilakukan oleh Choidkk (2010), dan Firdausillah (2011) menerapkan teknik sinkronisasi SyncML pada pengembangan *Mobile Repository System* pada perangkat bergerak sehingga dapat melakukan sinkronisasi kontak alamat dan telepon. Hasil penelitian menunjukkan bahwa perangkat bergerak mampu menyimpan kontak data yang disinkronisasikan dari server untuk kemudian disimpan di perangkat bergerak. Penelitian yang dilakukan oleh Skogstad tidak menggunakan *mobile database*.

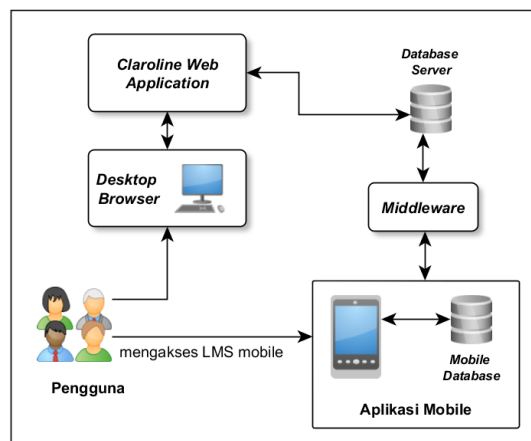
Selanjutnya Jotiphriya dan Shri (2013) juga melakukan penelitian mengenai sinkronisasi data. Jotiphriya menerapkan teknik sinkronisasi untuk mensinkronisasikan data mahasiswa dari basis data server ke basis data pada perangkat bergerak. Jotiphriya menggunakan Microsoft Synchronization Framework sebagai teknik sinkronisasinya dan menggunakan *mobile database* SQL Server Compact Edition.

Di samping penelitian mengenai sinkronisasi, terdapat pula beberapa penelitian tentang penerapan *mobile database* dalam pembelajaran. Misalnya Hanafidan Samsudin (2012) dan Iskandar (2010) membahas beberapa keunggulan dan kekurangan penerapan komputasi mobil pada pembelajaran. Kemudian penelitian Anaraki (2008) dan Godwin-Jones (2011) pada pembelajaran penerjemahan bahasa. Serta Kotecha dan Chajjed (2012) mencoba mengembangkan konsep model matematis dan web service untuk aplikasi mobil.

### 3. Metode Penelitian

#### 3.1 Rancangan sistem

Gambar 1 mendeskripsikan gambaran umum dari sistem yang akan dibangun dimana terdiri dari tiga komponen yaitu aplikasi mobile, middleware dan server berupa LMS berbasis web. Gambar tersebut mendeskripsikan keadaan dimana pengguna dapat mengakses LMS berbasis web melalui aplikasi mobile yang dapat berinteraksi dengan server melalui middleware.



Gambar 1. Rancangan Sistem

Berdasar pada rancangan tersebut, basis data server dan middleware terletak pada satu server yang sama dengan LMS. Kemudian aplikasi mobile dapat mengakses LMS melalui jaringan intranet. Pengguna dapat melakukan sinkronisasi data dari server ke aplikasi mobile melalui middleware. Data yang telah disinkronisasikan dari server akan disimpan pada mobile database yang terletak di perangkat bergerak.

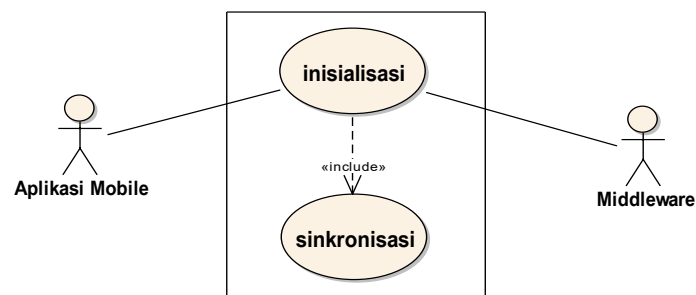
Penelitian ini akan memanfaatkan LMS berbasis web sebagai aplikasi pada server. LMS yang digunakan yaitu LMS Claroline dimana pada implementasinya hanya menggunakan beberapa modul yang ada pada LMS tersebut yaitu modul *course description*, *assignment*, *announcement* dan *agenda*. Gambaran sistem akan diimplementasikan pada studi kasus untuk pembelajaran pada mata kuliah English Basic. Skenario dalam penelitian ini

digambarkan dimana pengguna dalam hal ini adalah mahasiswa dapat mengakses LMS yang berisi mata kuliah English Basic melalui perangkat bergerak.

Sedangkan komponen *middleware* dikembangkan dengan tujuan untuk menyediakan layanan sinkronisasi data antara server yang dalam hal ini adalah LMS berbasis web dengan *client* yaitu aplikasi *mobile*. *Middleware* dibangun dengan mengembangkan fungsi-fungsi untuk mengatur proses sinkronisasi. Pengembangan *middleware* dilakukan dengan menggunakan bahasa PHP dan standar protokol sinkronisasi SyncML. Selain aplikasi server dan *middleware* akan dibangun pula aplikasi *mobile* yang merupakan aplikasi yang dikhususkan untuk diakses melalui perangkat bergerak. Aplikasi ini dilengkapi dengan basis data yang digunakan untuk menyimpan data hasil sinkronisasi dengan server dan basis data yang digunakan adalah SQLite. Aplikasi dibangun menggunakan bahasa HTML, CSS dan Javascript.

### 3.2 Use Case Proses Sinkronisasi

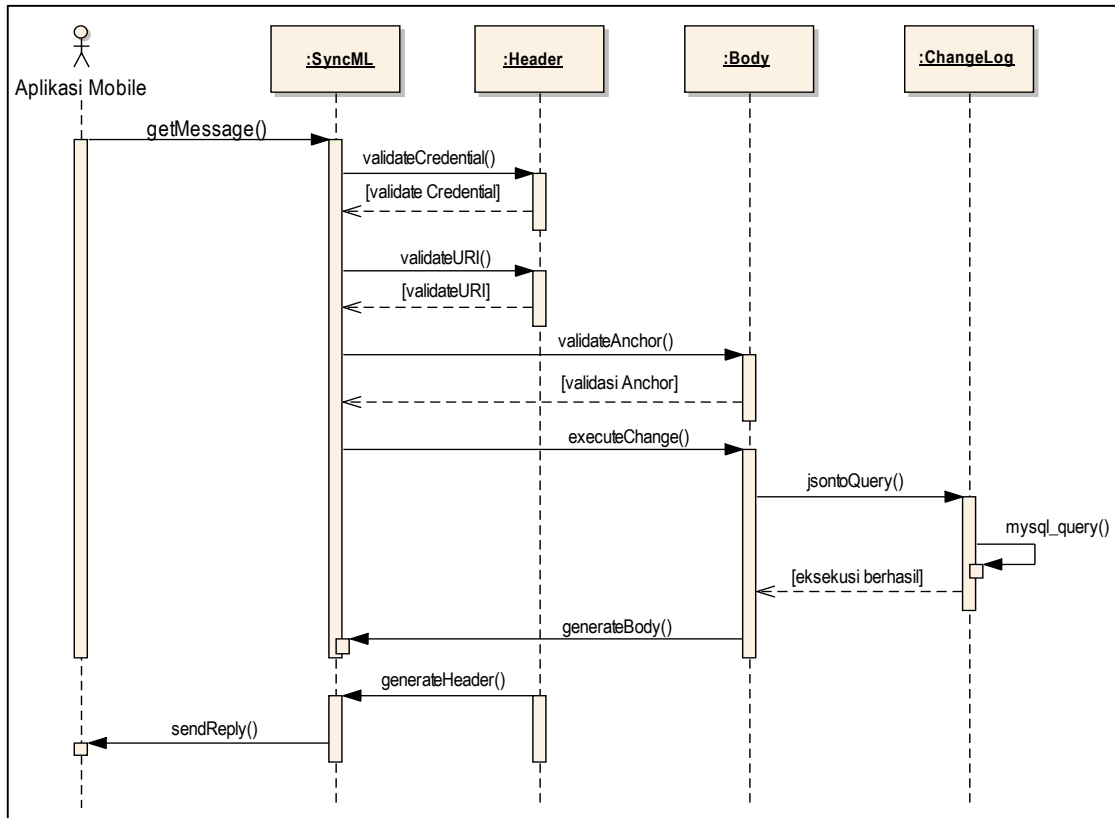
Gambaran mengenai proses sinkronisasi antara aplikasi *mobile* yang dibangun dengan *middleware* terdeskripsi pada Gambar 2. Aplikasi yang dibangun diharapkan dapat berinteraksi dengan melakukan proses sinkronisasi ke server melalui *middleware*. Proses sinkronisasi dilakukan dengan melalui proses inialisasi untuk menentukan mode sinkronisasi. Setelah proses inialisasi selesai maka proses sinkronisasi dilakukan oleh pihak *client* dan server.



Gambar 2. Diagram use case aktivitas sinkronisasi

### 3.3 SequenceDiagram untuk Proses Sinkronisasi

Deskripsi diagram sekuen yang ditunjukkan pada Gambar 4 difokuskan pada penggambaran pada proses yang terjadi antara *client* dengan *middleware* yaitu pada proses sinkronisasi yang merupakan proses pengiriman dan penerimaan pesan data.



**Gambar 3.** Diagram sekuen proses sinkronisasi

Proses sinkronisasi atau penerimaan dan pengiriman pesan data melibatkan aktor berupa aplikasi mobile yang berinteraksi dengan middleware dengan kelas yang teridentifikasi yaitu kelas SyncML, Header, Body dan changeLog. Proses penerimaan dan pengiriman pesan data dilakukan pada saat pengguna membuka aplikasi dan menekan tombol Sync. Kemudian kelas SyncML menanggapi dengan memanggil fungsi getMessage() untuk mem-parsing pesan.

Kelas SyncML memiliki dua bagian yaitu bagian header dan body. Bagian pesan header akan dikirim ke kelas Header sedangkan bagian pesan body dikirim ke kelas Body. Kelas Header akan menangani data credential dan URI dengan memanggil fungsi validateCredential() dan validateURI() dan kemudian memberikan nilai hasil validasinya. Kemudian kelas Body menangani validasi anchor dengan menjalankan fungsi validateAnchor() untuk menentukan mode sinkronisasi.

Setelah itu, objek Body akan memanggil objek changeLog untuk mengeksekusi pesan SyncML yang merupakan hasil modifikasi data dari client dan data tersebut kemudian dieksekusi di dalam basis data server. Terakhir, semua perubahan data yang dilakukan pada server sejak sinkronisasi terakhir akan di-generate dengan memproses fungsi generateBody(), generateHeader() dan mengembalikan pesan tersebut dengan memanggil fungsi sendReply().

### 3.4 Model Pesan Sinkronisasi

Pengembangan *middleware* dilakukan dengan pembuatan kode program untuk mengatur proses sinkronisasi dan merupakan kode paling penting dalam penelitian ini. Kode program yang berupa pesan SyncML memungkinkan proses sinkronisasi dapat dilakukan

oleh server dan *client*. Pengembangan pesan SyncML yang ditulis dengan XML memiliki dua jenis pesan utama yaitu pesan inisialisasi dan paket pesan sinkronisasi.

#### a. Pesan Inisialisasi

Pesan inisialisasi merupakan pesan yang dikirimkan sebelum paket pesan sinkronisasi dikirimkan. Pesan inisialisasi akan memproses autentikasi antara server dan *client*. Hasil pengiriman pesan inisialisasi tersebut berupa hasil validasi pesan inisialisasi yang dikirim ke server untuk menentukan mode sinkronisasi.

Pesan inisialisasi memiliki dua elemen penting yaitu elemen *<SyncHdr>* dan elemen *<SyncBody>*. Elemen *<SyncHdr>* ditentukan beberapa elemen yang disertakan pada bagian *<SyncHdr>* yaitu Session ID, MsgId, Target, Source dan Credential. Session ID: digunakan untuk mengindikasikan ID dari sesi sinkronisasi yang merupakan waktu kapan terjadinya sinkronisasi. MsgID untuk mengidentifikasi pesan dengan nilai "1" atau "2". Nilai "1" berarti pesan merupakan jenis pesan inisialisasi atau login, sedangkan nilai "2" merupakan pesan sinkronisasi. Target mengidentifikasi perangkat yang dituju. Source mengidentifikasi perangkat sumber dan Credential : untuk keperluan autentikasi.

Sedangkan elemen *<SyncBody>* terdapat tiga elemen yang perlu disertakan pada pesan inisialisasi yaitu CmdId yang digunakan untuk menentukan perintah sinkronisasi apakah berupa pesan inisialisasi atau untuk pesan sinkronisasi, Mode yaitu elemen mode akan menentukan jenis sinkronisasi yang dilakukan yaitu *slow synchronization* atau *fast synchronization* dan Anchor : elemen anchor digunakan sebagai acuan dalam melakukan sinkronisasi.

#### b. Pesan Sinkronisasi SyncML

Pesan sinkronisasi SyncML merupakan pesan penting dalam sinkronisasi karena di dalam pesan tersebut terdapat data yang akan disinkronisasikan. Pesan sinkronisasi SyncML juga memiliki dua elemen penting yaitu elemen *<SyncHdr>* dan elemen *<SyncBody>*. Seperti pada pesan inisialisasi bagian *header*, pesan sinkronisasi SyncML bagian *header* juga memiliki beberapa elemen yang disertakan yaitu seperti *sessionId*, *messageId*, *target*, *source*, dan *credential*. Sedangkan pada bagian *body*, pesan sinkronisasi terdiri dari *commandId*, mode sinkronisasi, *anchor* dan data utama berupa JSON data.

Kode program untuk pesan sinkronisasi yang dikirim dari *client* ke server ditunjukkan pada Gambar 4. Isi pesan pada bagian *header* dan *body* mirip dengan pesan inisialisasi. Perbedaan antara pesan inisialisasi dan pesan sinkronisasi ada pada bagian *body*, dimana terdapat data yang akan disinkronkan dari *client* ke server dan sebaliknya.

Data yang dikirim pada pesan sinkronisasi diformat dalam bentuk JSON Data. Dalam pesan tersebut terdapat data-data penambahan, perubahan atau penghapusan data pada tabel yang telah dilakukan oleh server. Gambar 5 menunjukkan contoh isi dari pesan sinkronisasi bagian data dalam bentuk JSON Data.

```

2 <SyncML>
3   <SyncHdr>
4     <SessionID>1</SessionID>
5     <MsgID>1</MsgID>
6     <Target><LocURI>http://sync.claroline.com</LocURI></Target>
7     <Source>
8       <LocURI>{"uuid":"ASDF09A8FA987FA098A",
9         "platform":"Android","model":"Galaxy Tab P6200",
10        "os_version":"4.1.2","utc_time":"UTC+7"}</LocURI>
11     </Source>
12     <Cred>{"username":"kaqfa","password":"123"}</Cred>
13   </SyncHdr>
14   <SyncBody>
15     <CmdID>2</CmdID> <!-- 1 = INIT; 2 = SYNC -->
16     <Mode>200</Mode> <!-- 200 = TWO_WAY_ALERT -->
17     <Anchor>
18       <Last>234</Last>
19       <Next>276</Next>
20     </Anchor>
21     <Data>
22       <!-- Data sinkronisasi dalam format JSON Data -->
23     </Data>
24   </SyncBody>
74 </SyncML>

```

Gambar 4. Pengisian pesan sinkronisasi

```

24 <Data> <!-- JSON messages -->
25   {
26     "insert": [
27       {
28         "name": "c_EN_course_description",
29         "cols": [
30           "id",
31           "category",
32           "title",
33           "content",
34           "lastEditDate",
35           "visibility"
36         ],
37         "vals": [
38           "13",
39           "9",
40           "Description",
41           "Course English Basic",
42           "2013-01-06 08:22:28",
43           "VISIBLE"
44         ]
45       }
46     ],
47   }
48 </Data>

```

Gambar 5. Data pada pesan sinkronisasi

#### 4. Hasil dan Pembahasan

Sub bab ini menjelaskan mengenai hasil pengujian dan pembahasannya. Pengujian terhadap aplikasi dilakukan dengan menguji performa guna mengetahui performa perangkat bergerak saat mengakses aplikasi yang dikembangkan. Pengujian dilakukan terhadap lima perangkat berupa dua tablet dan 3 smartphone dengan spesifikasi tercantum pada tabel 1. Tabel tersebut menguraikan spesifikasi berupa sistem operasi, prosesor dan memori yang dimiliki tiap perangkat yang digunakan pada pengujian.

Kolom kode pada Tabel 1 merupakan kode untuk mempermudah dalam pencantuman nama perangkat untuk pengujian. Samsung Galaxy Tab 7.0 dengan kode A memiliki spesifikasi tertinggi, dengan sistem operasi Android versi 4.0, prosesor 1.2 GHz dan dilengkapi dengan memori sebesar 1GB. Andromax Tab 7.0 juga memiliki spesifikasi perangkat yang hampir mirip, tetapi memiliki spesifikasi prosesor yang lebih kecil yaitu 1 GHz. Kemudian smartphone Sony Xperia Mini Pro merupakan smartphone dengan sistem operasi Android versi 2.3 atau GingerBread dan memiliki prosesor 1 GHz serta memori sebesar 1 GB.

Untuk Samsung Galaxy Tab 7.0 dan Sony Xperia mini menggunakan versi Android 4.0, sedangkan pada Samsung Galaxy Mini memiliki sistem operasi versi Android 2.3 atau biasa disebut dengan GingerBread. Smartphone dengan spesifikasi terendah yang digunakan untuk pengujian yaitu Samsung Galaxy Mini, yang memiliki Android versi 2.2 untuk sistem operasinya, prosesor dengan ukuran 600MHz dan memori sebesar 512 MB.

Pengujian untuk mengetahui performa aplikasi dilakukan dengan menghitung waktu untuk mengakses aplikasi setelah login dan pada saat pertama kali melakukan sinkronisasi penuh atau full synchronization yang dilakukan pada waktu yang bersamaan dan berbeda.

**Tabel 1.** Spesifikasi perangkat pengujian

Kode	Smartphone	Sistem Operasi	Prosesor	Memori
A	Samsung Galaxy Tab 7.0	Android 4.0	1.2 GHz DualCore	1 GB
B	Andromax Tab 7.0	Android 4.0	1 GHz	1 GB
C	Sony Xperia Mini SK17i	Android 2.3	1 GHz	512 MB
D	Samsung Galaxy Ace	Android 2.2	800 Mhz	768 MB
E	Samsung Galaxy Mini	Android 2.2	600 MHz	512 MB

Tabel 2 menunjukkan hasil pengujian untuk waktu dalam melakukan proses login pada berbagai perangkat. Dari tabel tersebut diketahui bahwa untuk rata-rata waktu yang digunakan pada proses login relatif tidak jauh berbeda yaitu berkisar antara 1300 ms atau sekira 1 detik. Hal ini dikarenakan pada saat proses login, data yang dikirimkan hanya berupa data akun pengguna yang tersimpan pada server.

**Tabel 2.** Pengujian waktu untuk login pada berbagai perangkat

Perangkat	A	B	C	D	E
Waktu (ms)	1356	1396	1393	1302	1312

Pengujian berikutnya merupakan pengujian terhadap perangkat untuk melakukan proses full synchronization dengan cara mengakses aplikasi mobile dan terhubung dengan internet untuk melakukan sinkronisasi dengan server. Tabel 6.3 menampilkan waktu yang dibutuhkan untuk berbagai perangkat dalam melakukan sinkronisasi yang dilakukan pada



waktu yang berbeda dengan ukuran file yang bervariasi. Pengujian sinkronisasi data dengan ukuran data kecil yaitu 13 KB dengan dilakukan dengan ukuran waktu yang relatif sama yaitu pada rentang waktu 2600 ms-2800ms atau sekitar 2 detik. Meski waktu untuk sinkronisasi hampir sama, tetapi untuk perangkat dengan spesifikasi terendah memiliki waktu paling lama untuk sinkronisasi data sebesar 13 KB, yaitu dengan ukuran waktu 2898 ms.

Kemudian pengujian sinkronisasi data dengan ukuran data terbesar pada pengujian yaitu sebesar 992 KB diakses oleh perangkat Samsung Galaxy Tab 7.0 dengan ukuran waktu selama 5233 ms atau sekitar 5 detik. Berbeda dengan perangkat Samsung Galaxy Mini yang memiliki spesifikasi terendah, dimana perangkat ini membutuhkan waktu paling lama untuk mensinkronisasikan data yaitu sekitar 9 detik.

Berdasar pada data yang ditampilkan pada tabel tersebut, maka kesimpulan yang didapat dari pengujian ini adalah semakin besar ukuran file maka semakin besar pula waktu yang dibutuhkan untuk melakukan sinkronisasi. Kebutuhan waktu untuk melakukan sinkronisasi juga bergantung pada spesifikasi perangkat dimana semakin tinggi spesifikasi perangkat tersebut, maka waktu yang dibutuhkan akan semakin kecil.

**Tabel 3.** Pengujian sinkronisasi data pada waktu berbeda

No.	Data	Perangkat				
		A	B	C	D	E
1.	13 KB	2693 ms	2701 ms	2717 ms	2753 ms	2898 ms
2.	50 KB	2704 ms	2730 ms	2760 ms	2799 ms	2942 ms
3.	120 KB	2803 ms	2843 ms	2898 ms	2980 ms	2991 ms
4.	210 KB	3012 ms	3134 ms	3342 ms	3442 ms	3532 ms
5.	373 KB	3251 ms	3297 ms	3417 ms	4012 ms	4443 ms
6.	552 KB	3818 ms	3512 ms	3912 ms	5214 ms	7069 ms
7.	686 KB	3978 ms	3998 ms	4124 ms	5133 ms	7165 ms
8.	782 KB	4132 ms	4093 ms	4522 ms	5334 ms	7834 ms
9.	850 KB	4398 ms	4501 ms	4932 ms	5423 ms	8632 ms
10.	992 KB	5233 ms	5423 ms	5692 ms	6009 ms	9423 ms

Selain melakukan pengujian sinkronisasi data pada berbagai perangkat dengan waktu yang berbeda, pengujian juga dilakukan pada waktu yang bersamaan. Hal ini berarti kelima perangkat akan mencoba untuk mengakses data pada saat waktu yang sama. Tabel 4 menunjukkan hasil pengujian berupa ukuran waktu untuk melakukan proses sinkronisasi pada waktu yang bersamaan.

**Tabel 4.** Pengujian sinkronisasi data dengan waktu yang bersamaan

No.	Data	Perangkat				
		A	B	C	D	E
1.	13 KB	2697 ms	2708 ms	2719 ms	2759 ms	2901 ms
2.	50 KB	2701 ms	2728 ms	2764 ms	2793 ms	2946 ms
3.	120 KB	2798 ms	2839 ms	2903 ms	2881 ms	3011 ms
4.	210 KB	3019 ms	3145 ms	3351 ms	3451 ms	3554 ms
5.	373 KB	3256 ms	3312 ms	3421 ms	4023 ms	4461 ms
6.	552 KB	3830 ms	3520 ms	3923 ms	5210 ms	7095 ms
7.	686 KB	3980 ms	3997 ms	4123 ms	5158 ms	7203 ms
8.	782 KB	4142 ms	4089 ms	4524 ms	5355 ms	7831 ms
9.	850 KB	4401 ms	4505 ms	4939 ms	5474 ms	8690 ms
10.	992 KB	5245 ms	5430 ms	5700 ms	6043 ms	9502 ms

Jika dibandingkan dengan waktu yang ditampilkan pada Tabel 3, terlihat bahwa sinkronisasi data dengan ukuran kecil sebesar 13 KB diakses oleh kelima perangkat dengan ukuran waktu yang relatif sama dengan ketika diakses pada waktu yang berbeda. Namun demikian, dengan bertambahnya ukuran data semakin terlihat perbedaan waktu dalam melakukan sinkronisasi data. Seperti terlihat pada ukuran waktu untuk mengakses data sebesar 992 KB, perangkat Samsung Galaxy Tab 7.0 mengakses data dengan ukuran waktu 5233 ms pada saat pengujian dengan waktu yang berbeda, dan dengan ukuran waktu sebesar 5245 ms untuk pengujian dengan waktu bersamaan. Hasil tersebut mengindikasikan bahwa ukuran waktu yang dibutuhkan bagi perangkat Samsung Galaxy Tab 7.0 baik pada sinkronisasi yang dilakukan pada waktu yang berbeda atau bersamaan, memiliki ukuran waktu akses yang hampir sama.

Kesimpulan tersebut juga berlaku bagi perangkat Samsung Galaxy Mini dimana tidak terdapat perbedaan mencolok untuk waktu akses pada proses sinkronisasi data. Pada saat melakukan pengujian sinkronisasi data dengan waktu yang berbeda, perangkat ini membutuhkan waktu 9423 ms, sedangkan pada saat proses dilakukan bersamaan perangkat memiliki waktu akses 9502 ms.

Dengan demikian, dapat disimpulkan bahwa kebutuhan waktu untuk sinkronisasi data pada berbagai perangkat baik dengan waktu yang berbeda atau bersamaan tidak memiliki perbedaan lama waktu yang signifikan. Kebutuhan waktu hanya bergantung pada spesifikasi perangkat dan ukuran data yang akan disinkronisasikan.

## 5. Kesimpulan

Setelah mengimplementasikan dan melakukan pengujian terhadap sistem yang dibangun, maka dapat diambil beberapa kesimpulan yaitu:

- 1 Telah dibangun aplikasi mobile yang digunakan untuk mengakses LMS Claroline. Aplikasi client yang dibangun telah dilengkapi dengan basis data lokal dan dapat terhubung dengan server melalui middleware untuk dapat melakukan sinkronisasi data. Middleware yang dibangun mampu menangani masalah sinkronisasi data menggunakan teknik sinkronisasi dengan berbasis pada protokol SyncML pada pengembangan aplikasi mobile sehingga aplikasi dapat mensinkronkan data dengan server.
- 2 Aplikasi mobile dilengkapi dengan basis data lokal SQLite sehingga dapat menyimpan data yang diperoleh dari hasil sinkronisasi dengan server.
- 3 Hasil yang didapat dari pengujian terhadap performa aplikasi adalah kebutuhan waktu untuk melakukan sinkronisasi data pada aplikasi bergantung pada spesifikasi perangkat yang digunakan dan besarnya ukuran data harus disinkronkan. Semakin tinggi spesifikasi perangkat yang dimiliki, maka semakin cepat waktu untuk proses sinkronisasi. Demikian pula dengan ukuran data dimana semakin besar ukuran data, maka waktu yang dibutuhkan untuk sinkronisasi data akan semakin lama.

## Daftar Pustaka

- Almrashdeh, I. A., Sahari, N., Zin, N.A.M. dan Alsmadi, M., 2011, Distance Learning Management System Requirments from Student's Perspective, *Journal of Theoretical and Applied Information Technology* 1,24, 17-27
- Alqahtani, A., S dan Goodwin, R., 2012, E-commerce Smartphone Application, *International Journal of Advanced Computer Science and Applications*, 3, 8, 54-59.

- Anaraki, F. B. 2008, A Flash-Based Mobile Learning System for English as a Second Language, *ABAC Journal*, 3, 28, 25-35
- Boyinbode, O. K. dan Akinyede, R. O., 2008, Mobile Learning: An Application of Mobile and Wireless Technologies in Nigerian Learning System, *International Journal of Computer Science and Network Security*, 11, 8, 386-392
- Choi, M.-Y., Cho, E.-A., Park, D.-H., Moon, C.-J., and Baik, D.-K, 2010, A Database Synchronization Algorithm for Mobile Devices. *Consumer Electronics, IEEE Transactions*, 56, 2, 392 –398.
- Firdausillah, F., 2011, Improved Data Availability in Mobile Database Using Loose-Coupled Push Synchronization, *Thesis*, Faculty of Information and Communication Technology, Universiti Teknikal Malaka, Malaka
- Godwin-Jones, R., 2011, Emerging Technologies Mobile Apps for Language Learning, *Language Learning & Technology* 15, 2, 2-11.
- Hanafi, H. F. dan Samsudin, K., 2012, Mobile Learning Environment System (MLES): The Case of Android-based Learning Application on Undergraduates' Learning, *International Journal of Advanced Computer Science and Applications*, 3, 3, 63-66
- Iskandar, D., 2010, Pengembangan Aplikasi Berbasis Teknologi *Mobile* untuk Pembelajaran, *Thesis*, Fakultas Teknik, Universitas Gadjah Mada, Yogyakarta
- Jothipriya, G. and Shri, M.L, 2013, Database Synchronization of Mobile-build by using Synchronization Framework, *International Journal of Engineering and Technology*, 5, 3, 2316-2321
- Jothipriya, G. and Shri, M.L, 2013, Database Synchronization of Mobile-build by using Synchronization Framework, *International Journal of Engineering and Technology*, 5, 3, 2316-2321
- Kotecha, K. Jain, D dan Chajjed, A., 2012, Mathematical Model of M-Learning Application for Android Based Mobile Devices using Web Services. *Prosiding Konferensi Internasional Komputer Intelezensi*, 1,7,202-212
- Rahman, K. A., Ghazali, S. A. dan Ismail, M. N., 2010, The Effectiveness of Learning Management System (LMS) Case Study at Open University Malaysia(OUM), Kota Bharu Campus, *Journal of Emerging Trends in Computing and Information Sciences*, 2,2, 73-79
- Sarrab, M., Elgamel, L. dan Aldabbas, H., 2012, Mobile Learning (M-Learning) and Educational Environments, *International Journal of Distributed and Parallel Systems*, 3,4, 31-38.