

Pengembangan Sistem Informasi Lembaga Bimbingan Belajar Al-Hasyimi

Muhammad Jauhar Fuadi¹, Retno Indah Rokhmawati², Mochamad Chandra Saputra³

Program Studi Sistem Informasi, Fakultas Ilmu Komputer, Universitas Brawijaya

Email: ¹jauharfuadi54@gmail.com, ²retnoindahr@ub.ac.id, ³andra@ub.ac.id

Abstrak

Lembaga Bimbingan Belajar (LBB) Al-Hasyimi merupakan lembaga pendidikan swasta yang menyediakan bimbingan belajar membaca Al-Qur'an dan Iqro' baik secara privat maupun kelompok. Dalam mengelola hasil belajar siswa, LBB Al-Hasyimi masih menggunakan cara konvensional. Setiap tutor harus mencatat hasil belajar siswanya dalam buku yang disebut dengan buku prestasi siswa. Kemudian buku ini akan direkap oleh sekretaris setiap satu bulan sekali. Selain itu tutor juga harus melakukan presensi setelah melakukan bimbingan. Dengan memanfaatkan teknologi informasi, permasalahan tersebut dapat dipermudah dengan menggunakan Sistem Informasi Lembaga Bimbingan Al-Hasyimi berbasis *web* yang dikembangkan dengan metode *waterfall*. Metode pengembangan ini meliputi analisis kebutuhan, perancangan, implementasi, dan pengujian. Pada tahap analisis kebutuhan dilakukan pemodelan proses bisnis *to-be* yang dibuat berdasarkan hasil wawancara dengan *stakeholder* dan analisis permasalahan. Hasil analisis kebutuhan divisualisasikan dalam diagram *use case*. Hasil perancangan sistem menghasilkan *sequence diagram*, *class diagram*, *physical data model*, *pseudocode algorithm*, *user interface*, dan *testing design*. Implementasi dari sistem informasi ini menggunakan *framework CodeIgniter* dengan menerapkan pola *Model-View-Controller*. Hasil *Validation Testing* yang dilakukan pada 6 kasus uji memperoleh presentasi 100% *valid*. Kemudian hasil *User Acceptance Testing (UAT)* yang dilakukan pada beberapa fitur utama, diantaranya menambahkan hasil belajar siswa, melihat hasil belajar, melihat absensi tutor, dan melakukan komplain memperoleh persentase rata-rata lebih dari 80%. Dengan hasil pengujian *UAT* tersebut, maka dapat disimpulkan bahwa Sistem Informasi Lembaga Bimbingan Belajar Al-Hasyimi dapat diterima oleh pengguna.

Kata kunci: *sistem informasi, waterfall, codeigniter, user acceptance testing, web.*

Abstract

Lembaga Bimbingan Belajar (LBB) Al-Hasyimi is a private educational institution that providing study guidance in reading the Holy Koran and Iqro' both by private and by group. In managing student's study result, LBB Al-Hasyimi still uses a conventional method. Every mentor has to note their student's study result in the book which called by "Buku Prestasi Siswa". Then this book would be recapped by secretary once in every month. In the other hand, the mentor had to do checking presence after guiding too. By using information technology, the problem can be simplified by using Lembaga Bimbingan Belajar Al-Hasyimi Information System based on website which developed by using waterfall method. This development method covered requirement analysis, design, implementation and testing. On the requirement analysis produced business process *to-be* that created based on the result of stakeholder interview and problem analysis. The result of the requirement analysis visualized in use case diagram. The result of the system design produced *sequence diagram*, *class diagram*, *physical data model*, *pseudocode algorithm*, *user interface*, and *testing design*. On its implementation, this system developed by using *CodeIgniter framework* by applying the *Model-View-Controller* pattern. The result of *Validation Testing* on 6 test case got as many as 100% *valid*. Then the result of *User Acceptance Testing (UAT)* on some features, there are adding a study result, seeing a study result, seeing a tutor absence, and complaining got average percentage more than 80%. With the result of *UAT* revealed that Lembaga Bimbingan Belajar Al-Hasyimi Information System already accepted by user.

Keywords: *information system, waterfall, codeigniter, user acceptance testing, website.*

1. PENDAHULUAN

Lembaga Bimbingan Belajar (LBB) Al-Hasyimi merupakan sebuah lembaga yang didirikan oleh santri Pesantren Mahasiswa Al-Hikam Malang. LBB Al-Hasyimi ini bertujuan untuk menyalurkan kemampuan para santri yang mana terdiri dari berbagai jurusan untuk dapat membagikan ilmunya kepada Masyarakat. Untuk saat ini LBB Al-Hasyimi menyediakan bimbingan secara privat maupun kelompok yang terdiri dari bimbingan baca quran dan iqro'. Untuk siswanya sendiri dari usia 7 tahun hingga dewasa. Saat ini siswa LBB Al-Hasyimi sudah menyebar di seluruh kota Malang baik dari tingkat SD hingga Universitas.

Hasil wawancara kepada beberapa pihak dari LBB Al-Hasyimi, saat ini LBB ini masih memiliki kendala dalam proses perekapan data hasil bimbingan siswa. Dalam proses pencatatan hasil bimbingan masih menggunakan buku prestasi siswa. Dimana tutor setiap satu bulan sekali harus menyerahkan data tersebut ke bagian sekretaris yang bertugas merekap data tersebut. Untuk pengumpulan buku prestasi siswa tadi tutor harus menuju ke kantor LBB Al-Hasyimi. Data tersebut untuk selanjutnya akan dimasukkan ke Microsoft Excel untuk direkap. Selain itu setiap kali setelah melakukan bimbingan, tutor harus melakukan absensi ke Kantor.

Dari uraian masalah di atas maka dibutuhkan sistem informasi yang dapat melakukan manajemen pada pengelolaan tutor untuk mengefektifkan proses bisnis yang ada. Dimana sistem informasi tersebut dapat membantu dalam proses manajemen data – data bimbingan belajar. Sehingga data dapat terkumpul dengan cepat dan memudahkan sekretaris dalam melakukan perekapan data. Selain itu, hal ini berguna untuk meningkatkan kualitas tutor dengan adanya feedback baik dari siswa maupun orang tua siswa yang dapat dikirimkan langsung melalui sistem informasi ini. Bagi LBB Al-Hasyimi dapat melakukan pengelolaan data yang efisien dan memonitor para tutor dengan intensif.

Waterfall merupakan model pengembangan perangkat lunak yang menggunakan pendekatan sequential dalam proses pengembangannya (Pressman, 2001). Metode ini sangat sederhana dan memiliki tahap – tahap yang sistematis serta dokumentasi yang baik. Sehingga hal ini dapat membantu dalam proses pengembangan pada

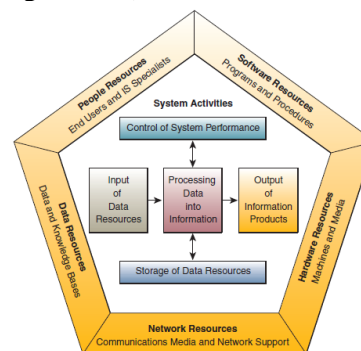
sistem informasi ini. Selain itu metode ini sangat cocok ketika kebutuhan dari sistem sudah diketahui di awal, sehingga dapat mempermudah dalam proses implementasi (Alshamrani & Bahattab, 2015). Maka dari itu, metode ini akan diterapkan pada penelitian ini.

Pada tahap akhir pengembangan, dilakukan pengujian terhadap sistem informasi dengan menggunakan 3 metode pengujian diantaranya *validation testing*, *compatibility testing*, dan *user acceptance testing*. *Validation testing* bertujuan untuk mengetahui apakah sistem yang dikembangkan sudah memenuhi persyaratan fungsional sehingga dapat dijalankan dengan baik dan sesuai harapan pengguna (Pressman, 2010). *Compatibility testing* bertujuan untuk mengetahui apakah sistem yang dikembangkan dengan menggunakan bahasa atau *script* tertentu dapat berjalan dengan baik ketika dijalankan di berbagai jenis dan versi *browser* (Pressman, 2010). *User acceptance testing* bertujuan untuk memastikan apakah sistem yang dikembangkan dapat diterima oleh pengguna (Hambling & Goethem, 2013).

2. LANDASAN KEPUSTAKAAN

2.1. Sistem Informasi

Menurut Obrien dan Marakas (2010) sistem adalah kumpulan elemen yang memiliki keterkaitan dan bekerja sama dalam menggapai tujuan sedangkan informasi adalah data yang sudah diproses sehingga bernilai manfaat bagi pengguna akhir. Maka dari itu yang dimaksud sistem informasi adalah sebuah pengorganisasian seperangkat manusia, komunikasi, data, jaringan, peraturan, prosedur, software dan hardware yang mana akan melakukan proses menyimpan, menerima, merubah dan menyebarkan informasi dalam sebuah organisasi (Obrien dan Marakas, 2010).



Gambar 1. Sistem Informasi

2.2. Metode Waterfall

Waterfall merupakan model pengembangan perangkat lunak yang menggunakan pendekatan sequential dalam proses pengembangannya (Pressman, 2001). Pada dasarnya proses pengembangan perangkat lunak terdiri dari spesifikasi, pengembangan, validasi, dan evolusi, namun pada model ini dipecah lagi menjadi fase proses yang terpisah diantaranya analisis kebutuhan, desain dan implementasi, pengujian, dan perawatan (Sommerville, 2011).

2.3. Proses Bisnis (Business Process)

Proses bisnis adalah kumpulan aktifitas yang saling berkoordinasi dalam sebuah organisasi untuk melakukan *input* dan menghasilkan *output* sesuai dengan tujuan (Weske, 2012). Aktifitas proses bisnis menjadi hal yang penting pada sebuah organisasi. Efektifitas dan efisiensi proses bisnis menjadi faktor penting dalam sebuah organisasi dalam menghadapi *competitor*. Teknologi informasi menjadi salah satu aspek penting yang harus dikembangkan dalam organisasi. Organisasi harus dapat menangani kecepatan arus informasi. Semakin cepat menerima informasi dan menyebarkan informasi, berguna untuk organisasi dalam mencapai tujuannya.

2.4. Unified Modeling Language (UML)

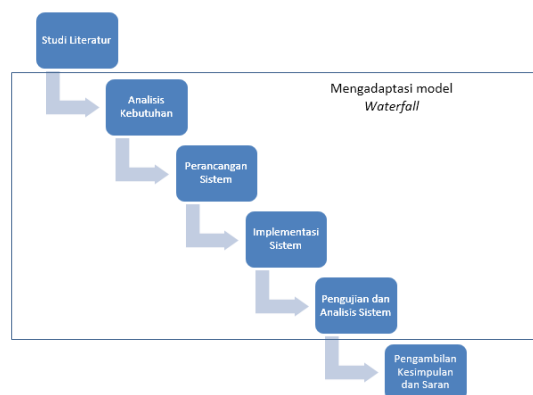
Unified Modeling Language (UML) adalah sebuah bahasa pemodelan standar untuk memvisualisasikan, merancang dan mengimplementasikan artefak dari analisa dan pengembangan perangkat lunak. *UML* juga digunakan untuk mendeskripsikan dan mendokumentasikan proses bisnis.

UML dapat diterapkan pada beragam domain aplikasi, seperti perbankan, keuangan, internet, e-commerce, dll). Selain itu juga dapat digunakan dengan semua metode pengembangan perangkat keras dan komponen utama untuk berbagai platform implementasi, seperti J2EE dan .NET. *UML* memiliki diagram-diagram yang digunakan untuk mendesain suatu sistem atau perangkat lunak, diantaranya yaitu *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, dan *Class Diagram*.

3. METODOLOGI PENELITIAN

Dalam melakukan tahap penelitian ini, penulis mengadopsi metode *waterfall* sebagai panduan dalam mengembangkan sistem

informasi LBB Al-Hasyimi. Pada Gambar 2 merupakan ilustrasi dari alur metodologi penelitian yang dilakukan penulis.



Gambar 2. Alur Penelitian

Studi literatur bertujuan untuk mempelajari serta mencari referensi lebih dalam untuk mendukung pengerjaan penelitian ini.

Analisis kebutuhan baik fungsional maupun non-fungsional serta menentukan siapa saja yang menjadi aktor dalam sistem ini.

Perancangan dilakukan dengan membuat gambaran sistem yang akan dikembangkan dengan menggunakan metode *Unified Model Language (UML)*.

Tahap implementasi mengacu kepada hasil perancangan yang sudah dibuat sebelumnya. Implementasi sistem menggunakan *framework Codeigniter*, dimana *framework* ini menggunakan design pattern *MVC*.

Pada tahap ini sistem yang sudah jadi akan diuji dengan menggunakan pengujian *black-box*. Pengujian ini berfokus pada pengujian kebutuhan sistem.

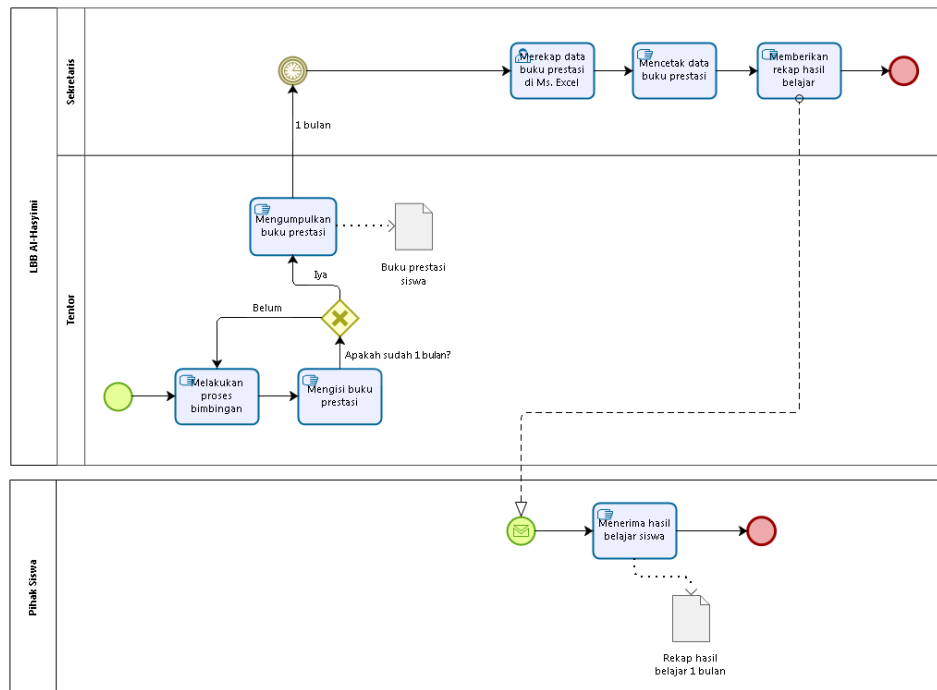
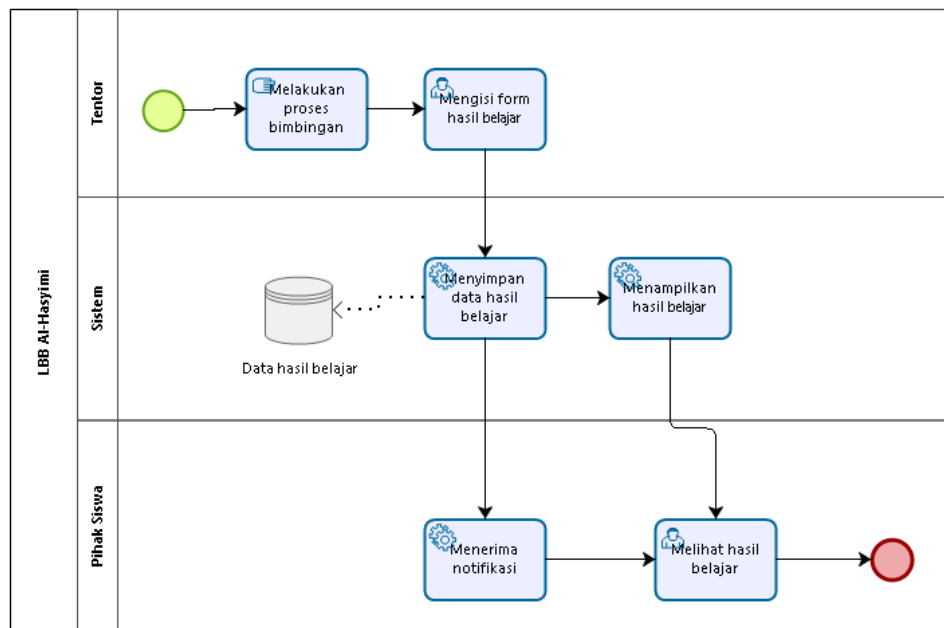
Setelah seluruh tahapan selesai dikerjakan, maka tahap terakhir yaitu pengambilan kesimpulan dan saran. Kesimpulan berisi hasil pengujian dan analisis sistem yang dikembangkan.

4. HASIL DAN PEMBAHASAN

Metode *waterfall* diterapkan dalam pengembangan sistem informasi, mulai dari analisis kebutuhan, perancangan, implementasi, serta pengujian.

4.1. Analisis Kebutuhan

Aktivitas proses bisnis dibuat menggunakan *Business Process Model and Notation (BPMN)* untuk memvisualisasikannya.

Gambar 3. Proses Bisnis *As-Is* Mengelola Hasil BelajarGambar 4. Proses Bisnis *To-Be* Mengelola Hasil Belajar

Gambar 3 merupakan proses bisnis saat ini, dimana tentor harus mencatat hasil belajar pada Buku Prestasi Siswa untuk kemudian dikumpulkan ke sekretaris satu bulan sekali. Kemudian sekretaris merekapnya dan memberikan laporan hasil belajar ke pihak siswa. Dengan adanya bantuan sistem informasi

pada Gambar 4, tentor hanya perlu memasukkan hasil belajarnya melalui sistem informasi LBB Al-Hasyimi. Kemudian sistem akan menyimpan dan mengirimkannya ke pihak siswa.

Dari analisis yang dilakukan pada proses bisnis *to-be* diatas. Maka diperoleh identifikasi aktor yang dijelaskan pada tabel 1.

Tabel 1. Identifikasi Aktor

Aktor	Peran
Sekretaris	Orang yang mengelola pengguna dan absensi tentor
Tentor	Orang yang mengelola hasil belajar siswa
Siswa	Orang yang menerima hasil belajar siswa
Orang Tua	Orang yang menerima hasil belajar siswa

Kemudian setelah dilakukan identifikasi proses bisnis *to-be* dan aktor, selanjutnya akan dilakukan analisis kebutuhan fungsional dan non-fungsional. Diperoleh 10 kebutuhan fungsional dan 3 kebutuhan non-fungsional. Kemudian pada spesifikasi kebutuhan yang didapat dari analisis kebutuhan dan identifikasi fitur, diperoleh 17 kebutuhan fungsional dan 1 kebutuhan non-fungsional. Terakhir, *diagram use case* untuk mengetahui interaksi antara aktor dan sistem. Pada Gambar 5 merupakan *diagram use case* dari sistem yang dikembangkan.



Gambar 5. Use Case Diagram

Setelah memperoleh *use case diagram*, selanjutnya akan dirancang bagaimana *scenario use case* dalam proses menambahkan hasil

belajar siswa dan melihat hasil belajar. Penjelasan *use case scenario* tersebut akan dijelaskan pada Tabel 2 dan Tabel 3. Tabel 2 merupakan merupakan *use case scenario* ketika tentor akan melakukan penambahan hasil belajar siswa pada sistem informasi LBB AI-Hasyimi.

Tabel 2. Use Case Scenario_Menambahkan Hasil Belajar

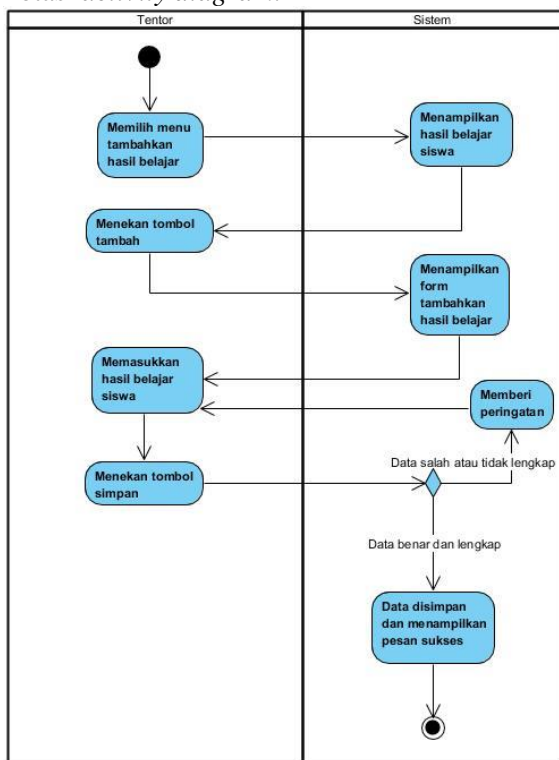
Item	Deskripsi
Use Case Code	UC-005
Use Case Name	Menambahkan hasil belajar
Actor	Tentor
Brief Description	Tentor menambahkan data hasil belajar siswa setelah melakukan bimbingan belajar
Pre-condition	Status login berhasil sebagai tentor
Basic Flow	<ol style="list-style-type: none"> 1. Tentor memilih menu tambahkan hasil belajar pada pilihan menu yang sudah disediakan 2. Sistem menampilkan <i>form</i> hasil belajar siswa yang harus diisi oleh aktor tentor 3. Tentor menambahkan hasil belajar siswa dengan mengisi <i>form</i> hasil belajar dengan lengkap dan benar 4. Tentor memilih fungsi simpan untuk menyimpan data hasil belajar siswa 5. Sistem memberikan pesan kepada tentor data berhasil disimpan
Alternatif Flow	<ol style="list-style-type: none"> 1. Jika data yang dimasukkan tidak lengkap maka sistem akan memberi notifikasi atau peringatan dan kembali ke langkah 1.
Post-Condition	Data disimpan dan dikirim oleh sistem

Pada Tabel 3 merupakan *use case scenario* melihat hasil belajar siswa. Aktor dari pihak siswa yang terdiri dari orang tua dan siswa dapat melihat hasil belajar siswa dengan langkah-langkah seperti yang dijelaskan pada *use case scenario* melihat hasil belajar berikut.

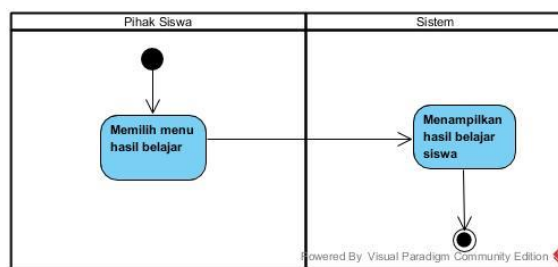
Tabel 3. *Use Case Scenario* Melihat Hasil Belajar

Item	Deskripsi
<i>Use Case Code</i>	UC-006
<i>Use Case Name</i>	Melihat Hasil Belajar Siswa
<i>Actor</i>	Siswa, Orang Tua
<i>Brief Description</i>	Siswa dan orang tua dapat melihat hasil belajar siswa
<i>Pre-condition</i>	Status login berhasil sebagai siswa atau orang tua
<i>Basic Flow</i>	<ol style="list-style-type: none"> 1. Siswa atau orang tua memilih menu hasil belajar 2. Sistem menampilkan informasi hasil belajar siswa
<i>Alternatif Flow</i>	-
<i>Post-condition</i>	Data hasil belajar siswa ditampilkan oleh sistem

Kemudian *activity diagram* akan menggambarkan bagaimana alur kerja dari *scenario use case* tersebut dengan menggunakan notasi *activity diagram*.

Gambar 6. *Activity Diagram* Menambahkan Hasil Belajar

Pada Gambar 6 menjelaskan tentang *activity diagram* memasukkan hasil belajar siswa yang terdiri dari *line* tentor dan sistem. Alur dimulai ketika tentor sudah diasumsikan berstatus login. Kemudian tentor memilih fungsi tambahkan hasil belajar dan mengisi form tersebut dengan lengkap. Terakhir, tentor memilih fungsi tombol kirim.

Gambar 7. *Activity Diagram* Melihat Hasil Belajar

Pada Gambar 7 menjelaskan tentang *activity diagram* melihat hasil belajar siswa yang terdiri dari *line* pihak siswa dan sistem. Alur dimulai ketika pihak siswa sudah diasumsikan berstatus login. Kemudian pihak siswa memilih menu hasil belajar dan sistem menampilkan hasil belajar tersebut.

4.2. Perancangan

Sequence diagram digunakan untuk melakukan visualisasi hubungan antara entitas aktor, entitas boundary, objek control, dan objek model dalam proses memenuhi kebutuhan pengguna.

Berdasarkan *sequence diagram* diatas, akan dibuat diagram kelas model dan diagram kelas controller, serta physical data model untuk menggambarkan desain *database* sistem.

Perancangan algoritma *pseudocode* dibuat sebagai dasar dalam membuat kode program. Pada Tabel 4 merupakan algoritma *pseudocode* memasukkan hasil belajar siswa. Kemudian Tabel 5 merupakan algoritma *pseudocode* melihat hasil belajar.

Tabel 4. Algoritma *Pseudocode* Menambahkan Hasil Belajar

No	Pseudeucode
1	Mulai
2	Mengkonversi data hasil belajar menjadi object
3	Memasukkan data ke tabel hasil belajar setelah dilakukan validasi form
4	

No	Pseudeucode
5	Jika tidak lengkap atau salah, maka muncul peringatan data harus diisi dengan benar
6	Jika data diisi dengan benar, maka data berhasil ditambahkan
7	Kembali kehalaman hasil belajar Selesai

Tabel 5. Algoritma *Pseudocode* Melihat Hasil Belajar

No	Pseudeucode
1	Mulai
2	Mengambil data hasil belajar berdasarkan id pengguna yang login
3	Melakukan pengecekan apakah terdapat data hasil belajar pada id tersebut
4	Jika data hasil belajar masih kosong, maka akan memberikan pesan data hasil belajar belum ada
5	Jika terdapat data hasil belajar, maka akan menampilkan data hasil belajar
6	Selesai

Perancangan *user interface* dibuat sebagai dasar dalam implentasi *user interface* sistem. Pada Gambar 8 merupakan rancangan *user interface* halaman memasukkan hasil belajar pada tentor. Kemudian setelah dipilih fungsi tombol tambah, maka akan menampilkan form hasil belajar yang dijelaskan pada Gambar 9.

Gambar 8. *User Interface Design* Halaman Hasil Belajar

Gambar 9. *User Interface Design Form* Hasil Belajar

Pada Gambar 10 merupakan perancangan *user interface* halaman hasil belajar siswa yang dapat diakses oleh pihak siswa. Bedanya dengan halaman hasil belajar siswa pada tentor yaitu tidak terdapat tombol tambah pada halaman hasil belajar pihak siswa ini.

Gambar 10. *User Interface Design* Halaman Hasil Belajar Pihak Siswa

4.3. Implementasi

Implementasi dibuat berdasarkan hasil pada tahap perancangan. Berikut pada Gambar 11 merupakan hasil implementasi dari *user interface* halaman hasil belajar pada tentor. Gambar 12 merupakan hasil implementasi dari *user interface form* hasil belajar yang harus diisi oleh tentor. Dan Gambar 13 merupakan hasil implementasi dari *user interface* halaman hasil belajar pada pihak siswa.

Gambar 11. *User Interface* Halaman Hasil Belajar

Menambahkan Hasil Belajar

Siswa

Tanggal

Jilid/Juz

Halaman/Ayat

Nilai

Catatan

Simpan

Keterangan Nilai
 Nilai di isi menggunakan huruf:
 - Nilai A (Sangat Baik)
 - Nilai B (Baik)
 - Nilai C (Cukup)
 - Nilai D (Kurang)
 - Nilai E (Sangat Kurang)

Gambar 12. User Interface Form Hasil Belajar

Hasil Belajar Siswa

No	Hari/Tanggal	Pukul	Jilid/Juz	Halaman/Ayat	Tentor	Nilai	Catatan
1	2018-06-20	14:00	Juz 10	Ayat 10	Ismu	A	tingkatkan
2	2018-07-30	14:00	Juz 10	Ayat 10	Ismu	A	Tingkatkan belajar

Gambar 13. User Interface Design Halaman Hasil Belajar Pihak Siswa

4.4. Hasil Pengujian

Hasil pengujian validasi pada beberapa fungsi utama diantaranya menambahkan hasil belajar, melihat hasil belajar, melihat absensi tentor, dan melakukan komplain menunjukkan hasil pengujian 100%. Hasil tersebut menunjukkan bahwa fungsi-fungsi tersebut dapat berjalan dengan baik.

Selain itu pada pengujian compability menggunakan software SortSite menunjukkan sistem dapat dijalankan diberbagai browser dengan versi yang berbeda-beda, namun masih terdapat beberapa *major* dan *minor issue* pada beberapa browser.

Hasil *User Acceptance Testing (UAT)* yang dilakukan pada beberapa fitur utama, diantaranya menambahkan hasil belajar siswa, melihat hasil belajar, melihat absensi tentor, dan melakukan komplain memperoleh presentase rata-rata lebih dari 80%. Dengan hasil pengujian

UAT tersebut, maka dapat disimpulkan bahwa Sistem Informasi Lembaga Bimbingan Belajar Al-Hasyimi dapat diterima oleh pengguna.

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan penelitian yang dilakukan, maka dapat disimpulkan bahwa hasil dari pemodelan proses bisnis usulan memperoleh 3 proses bisnis usulan. Kemudian dilakukan analisis kebutuhan fungsional sistem, dan didapatkan 17 kebutuhan fungsional dan 1 kebutuhan non-fungsional sistem.

Perancangan sistem menghasilkan *sequence diagram*, *class diagram*, *algoritma pseudocode*, *user interface*, dan perancangan pengujian. Hasil dari perancangan diimplementasikan dalam kode program berbasis website dengan menggunakan

framework CodeIgniter.

Hasil pengujian sistem menggunakan *validation testing* menunjukkan sistem 100% *valid*. Selanjutnya pengujian komparabilitas menunjukkan bahwa sistem dapat berjalan di berbagai jenis *browser*. Terakhir, pengujian *UAT* menunjukkan bahwa rata-rata pengguna sangat setuju dengan sistem yang dikembangkan, sehingga sistem dapat diterima oleh pengguna

5.2 Saran

Diperlukannya pengembangan lebih lanjut dari aplikasi ini dengan menambahkan fitur pendaftaran online, pembayaran online, dan fitur penjadwalan sehingga dapat lebih terkontrol dengan baik proses bimbingan ini.

Diperlukannya pengembangan user interface yang lebih baik untuk meningkatkan user experience dan usability pengguna, sehingga pengguna dapat lebih mudah dan nyaman menggunakan sistem ini.

6. DAFTAR PUSTAKA

- Alshamrani, A., dan Bahattab, A. 2015. *A Comparison Between Three SDLC Models Waterfall Model, Spiral Model, and Incremental/Iterative Model*. [online] Tersedia di: <<http://IJCSI.org>> [Diakses 12 Februari 2018].
- Jannah, W., Astuti, Indah F., & Maharani, S., 2015. Rancang bangun sistem informasi Bimbingan Belajar berbasis Web (Studi Kasus: Lembaga Bimbingan Belajar TADICA). [online] Tersedia di: <<http://e-journals.unmul.ac.id/index.php/JIM/article/view/20>> [Diakses 2 Februari 2018].
- Laudon, Kenneth C. & Laudon, Jane P. 2014. *Management Information System: Managing the digital firm* 13 th. [e-book] England: Pearson. Tersedia di: <<http://dinus.ac.id>> [Diakses 12 Februari 2018].
- O'Brien, James A., & Marakas, George M. 2010. *Management Information Systems* 10 th. [e-book] US : McGraw-Hill. Tersedia di: <<http://fumblog.um.ac.ir>> [Diakses 6 Februari 2018].
- Pressman, Roger S., 2001. *Software Engineering : A Practitioner's Approach* — 5th ed. [e-book] US : McGraw-Hill. Tersedia di: <<https://downloadnema.com>> [Diakses 27 September 2016].
- Rosing, Mark V., White, S., Cummins, F., & Man, Henk D. 2015. *Business Process Model and Notation*. [pdf] Tersedia di: <http://www.omg.org/news/whitepapers/Business_Process_Model_and_Notation.pdf> [Diakses 13 Februari 2018].
- Sommerville, I., 2011. *Software Engineering* 9 th. [e-book] U.S.A : Addison-Wesley. Tersedia di: <<https://edisciplinas.usp.br>> [Diakses 19 Desember 2016].
- Weske, M., 2007. *Business Process Management Concept, Languages, Architectures*. [e-book] New York: Springer. Tersedia di: <<http://manajemen-pemasaran.com>> [Diakses 21 Desember 2016].