Aplikasi Pengolahan Data Bimbel Rumah Belajar Solusi Berbasis Java Desktop

Enjel Lia, *Sistem Informasi, Universitas Katolik Musi Charitas,* dan Stefanus Setyo Wibagso, *Sistem Informasi, Universitas Katolik Musi Charitas*

***Abstract*— Rumah Belajar Solusi merupakan tempat bimbingan belajar yang melatih anak untuk mengembangkan pelajaran dalam bidang akademis siswa yaitu Matematika dan Ilmu Pengetahuan Alam (IPA). Pada binbel ini ditemukan permasalahan bahwa dalam proses pengolahan data masih dilakuakn secara manual. Maka dari itu dibuatlah aplikasi pengolahan data bimbel Rumah Belajar Solusi berbasis desktop untuk membantu *admin* dapat lebih cepat dan mudah dalam proses pengolahan data-data bimbel. Pembanguann aplikasi ini menggunakan metodologi *extreme programming* (XP). Analisis permasalahan menggunakan *PIECES*, berbasis *localhost*, dengan menggunakan bahasa pemrograman Java dan *database* Mysql. Penerapan aplikasi pengolahan ini diharapkan dapat membantu bagian *admin* dalam memproses pengolahan data bimbel.**

***Index Terms*— Rumah Belajar Solusi, aplikasi pengolahan data bimbel, metodologi *extreme programming*, *PIECES*, Java, Mysql.**

I. PENDAHULUAN

Rumah Belajar Solusi merupakan tempat bimbingan belajar yang melatih anak untuk mengembangkan pelajaran dalam bidang akademis siswa yaitu Matematika dan Ilmu Pengetahuan Alam (IPA). Rumah Belajar Solusi didirikan pada tanggal 18 Juli 2010 oleh Bapak Lewy Lukas. S.E., M.Pd. yang merupakan lulusan Pasca Sarjana Pendidikan Matematika Universitas Sriwijaya Palembang.

Beliau memiliki kompetensi yang memadai dibidang MIPA, maka beliau berinisiatif untuk mengajar privat MIPA. Kemudian tidak lama itu, beliau memutuskan untuk membuat sebuah tempat bimbingan belajar yang dinamakan Rumah Belajar Solusi yang merupakan awal usahanya.

Dahulu bimbingan belajar Rumah Belajar Solusi berada di Jalan Dempo Dalam 1090/III sebelum akhirnya pindah lokasi di Jalan kebun Manggis, Kepandean Baru dikarenakan kapasitas tempat yang tidak mencukupi. Sekarang jumlah siswa di Rumah Belajar Solusi telah mencapai 150 siswa diantaranya terdapat kelas SD, SMP, maupun SMA.

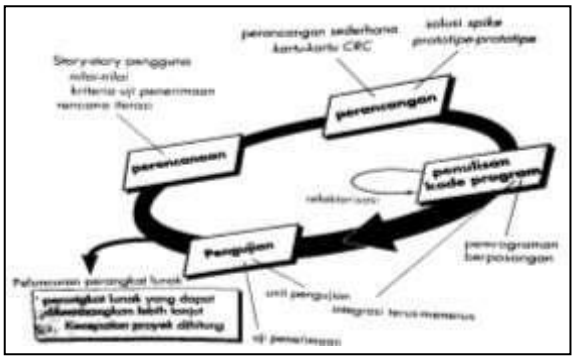
II. KEGIATAN KERJA PRAKTEK

Dalam bimbel Rumah Belajar Solusi memiliki bagain administrasi dan staff pengajar. Bagian unit kerja yang akan diteliti oleh penulis adalah bagian administrasi yang melakukan berbagai hal diantaranya menginput pendaftaran siswa baru, pengelolaan data siswa, menyusun jadwal les, mencatat pembayaran siswa tiap bulan, serta membuat laporan pembayaran.

III. METODOLOGI PENGEMBANGAN SISTEM

Metodologi adalah kesatuan metode-metode, prosedur-prosedur, konsep-konsep pekerjaan, aturan-aturan dan postulat-postulat yang digunakan oleh suatu ilmu pengetahuan, seni atau disiplin lainnya[4]. Metodologi pengembangan sistem berarti metode-metode, prosedur-prosedur, konsep-konsep pekerjaan, aturan-aturan dan postulat-postulat yang digunakan untuk mengembang suatu sistem informasi.

Metodologi pengembangan sistem yang akan digunakan adalah Metode *Extreme Programming* (XP). Metode XP dicetuskan oleh Kent Beck dan berkembang sebagai jawaban atas masalah-masalah yang ditimbulkan pada proses lamanya pengembangan perangkat lunak dengan metode pengembangan tradisional dikarenakan memiliki kecenderungan membutuhkan waktu yag lama untuk setiap tahapannya.



Gambar 1 Metode *Extreme Programming*
Sumber : Pressman, 2010

Terdapat 4 tahapan dalam pengembangan perangkat lunak dengan menggunakan XP sebagai berikut :

1. *Planning* (Perencanaan)
   Tahap awal dimana pengembang membuat perencanaan untuk menggambarkan hasil keluaran dan fungsional dari aplikasi. Kegiatan perencanaan meliputi identifikasi permasalahan, menganalisa kebutuhan sampai dengan

penetapan jadwal pelaksanaan pembangunan. Informasi mengenai kebutuhan-kebutuhan tersebut didapat dari dokumentasi *user story*.

- Identifikasi Permasalahan

Suatu permasalahan yang terjadi pada sistem yang lama dapat memberikan dampak yang buruk bagi sistem yang digunakan. Oleh karena itu, diperlukannya identifikasi permasalahan pada sistem yang berjalan sehingga dapat terpenuhi sesuai dengan keinginan pengguna.

Untuk mengidentifikasi maslaah yang ada, maka perlu dilakukan analisis terhadap kinerja, informasi, ekonomi, keamanan, efisiensi, dan pelayanan. Aspek ini dikenal dengan analisis *PIECES*. Selain menggunakan metode analisis *PIECES*, digunakan metode analisis sebab akibat, sehingga dari hasil analisis tersebut diharapkan menjadi landasan dalam membuat aplikasi yang dapat membantu bagian administrasi dalam mengolah data bimbel Rumah Belajar Solusi.

Tabel 1 Analisis Sebab Akibat

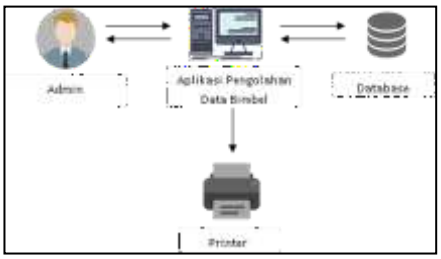
No	Kriteria	Permasalahan	Sebab	Akibat
1	*Performance* (Kinerja)	Pembentan informasi yang kurang cepat saat suatu informasi dibutuhkan	Data yang dicatat bersifat manual	Memakan banyak waktu dalam mencari informasi yang dibutuhkan
2	*Information* (Informasi)	Pencatatan data yang secara manual kemungkinan dapat terjadi salah penulisan	Pencatatan data yang ditulis secara manual	Informasi yang didapatkan kurang akurat
3	*Economy* (Ekonomi)	Masih menggunakan pena, kertas, dan map	Pencatatan data menggunakan pena, kertas, dan map tanpa menggunakan bantuan sistem	Pemborosan biaya operasional untuk membeli peralatan tulis guna mencatat data
4	*Control* (Kontrol)	Belum ada sistem untuk menyimpan data bimbel	Kontrol dokumen yang sangat kurang	Data bimbel dapat hilang dan tidak tersusun dengan rapi bahkan rusak
5	*Efficiency* (Efisiensi)	Sumber daya yang digunakan lebih banyak	Masih melakukan semua kegiatan secara manual	Pemborosan biaya, waktu, dan peralatan terlebih jika terjadi kesalahan dalam pencatatan dan pencarian data
6	Service (Pelayanan)	Pelayanan pembayaran membutuhkan waktu yang lama dari pencarian data siswa hingga pembuatan [illegible]	Data yang dicatat bersifat manual	Menimbulkan antrian dengan waktu tunggu pelayanan yang tidak sebentar

2. *Design* (Desain)

Tahap desain merupakan bagian dari perancangan aplikasi yang sesuai dengan kebutuhan dari penggunanya yaitu berdasarkan pada fase *planning*. Pada fase ini dilakukan kegiatan pemodelan sistem (*Use Case Diagram*, *Class Diagram*, *Activity Diagram*) dengan menggunakan bahasa pemodelan UML (*Unified Model* Diagram).

- Perancangan Arsitektur Sistem

Arsitektur dari sistem secara keseluruhan yang akan dibuat pada data bimbel Rumah Belajar Solusi dapat dilihat pada Gambar 2 :



Gambar 2 Arsitektur Sistem yang Diusulkan

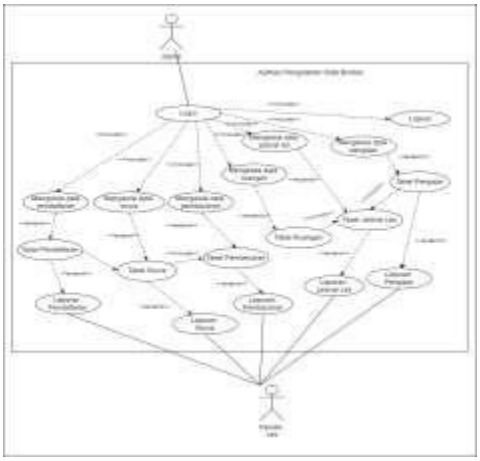
Berdasarkan Gambar 2 menjelaskan bahwa admin menginput data-data bimbel ke aplikasi pengolahan data bimbel dan tersimpan di *database*. *Database* yang telah disimpan dapat ditampilkan lagi dalam aplikasi dan dari aplikasi dapat memberikan informasi ke admin dan juga dapat dicetak dalam bentuk laporan melalui printer.

- Pemodelan Proses Sistem

UML (*Unified Modeling Language*) adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan, dan membangun perangkat lunak [3].

a) Use Case Diagram

Use Case Diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat[6]. *Use case diagram* yang diusulkan dapat dilihat pada Gambar 3.

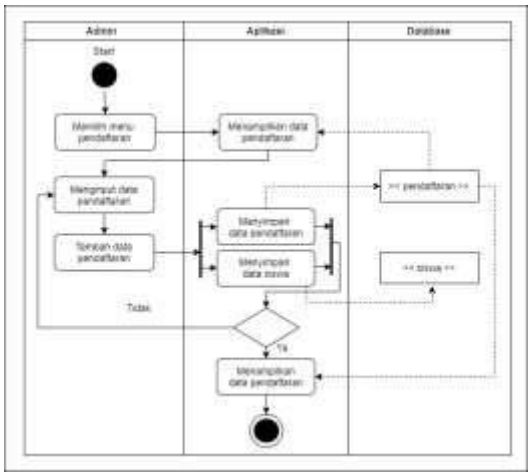


Gambar 3 *Use Case Diagram* yang Diusulkan

Berdasarkan Gambar 3.3 di atas, hanya ada 1 aktor dalam *use case diagram* yaitu admin yang menggunakan aplikasi. *Admin* dapat mengelola semua data yaitu data pendaftaran, data siswa, data pengajar, data ruangan, data jadwal les, dan data pembayaran serta *logout* yang dimana mengelola semua data tersebut harus melakukan login terlebih dahulu.

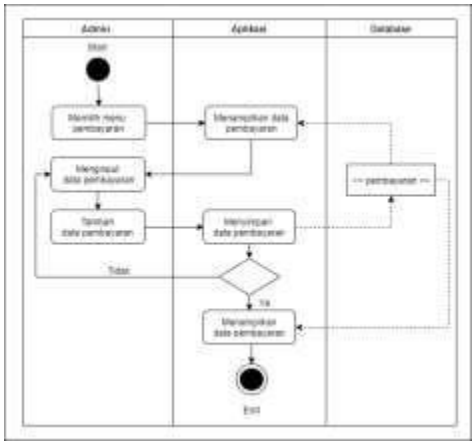
b) Activity Diagram

Diagram aktivitas atau a*ctivity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak[6].



Gambar 4 *Activity Diagram* Tambah Pendaftaran

Pada *activity diagram* aktivitas pertama yang dilakukan admin dalam *activity diagram* tambah data pendaftaran adalah memilih menu data pendaftaran dan *database* akan manggil data pendaftaran, setelah itu aplikasi akan menampilkan seluruh data pendaftaran. Setelah tampil seluruh data, admin dapat langsung menginput data pendaftaran dan megklik tombol tambah pada data pendaftaran. Setelah selesai menginput maka sistem akan menyimpan data tersebut dan otomatis akan langsung menyimpan ke *database* pendaftaran dan *database* siswa. Apabila data yang diinput benar maka aplikasi akan menmapilkan data pendaftaran dan apabila data salah maka akan kembali ke form input data pendaftaran.

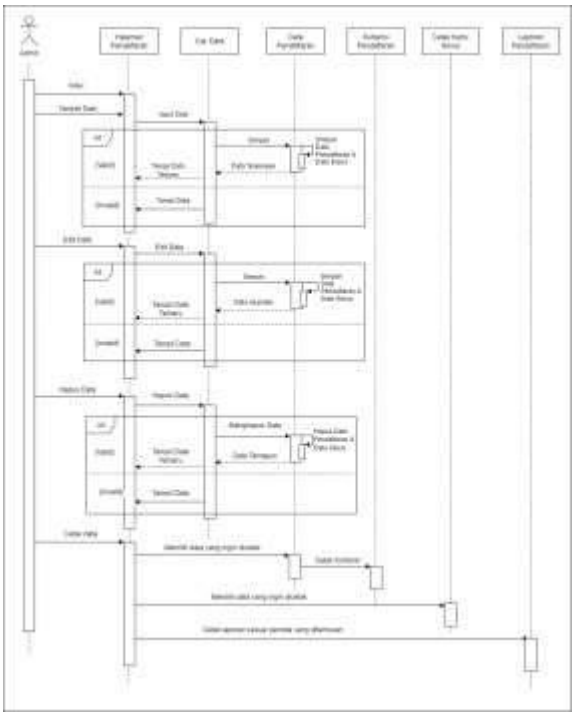


Gambar 5 *Activity Diagram* Tambah Pembayaran

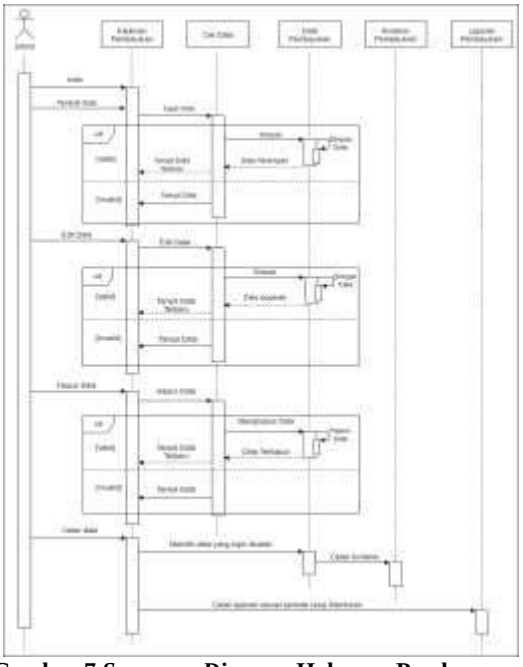
Pada *activity diagram* aktivitas pertama yang dilakukan admin dalam *activity diagram* tambah data pembayaran adalah memilih menu data pembayaran dan *database* akan manggil data pembayaran, setelah itu aplikasi akan menampilkan seluruh data pembayaran. Setelah tampil seluruh data, admin dapat langsung menginput data pembayaran dan mengklik tombol tambah pada data pembayaran. Setelah selesai menginput maka sistem akan menyimpan data tersebut dan otomatis juga akan tersimpan ke *database* pembayaran. Apabila data yang diinput benar maka akan langsung tampil data pembayaran ke dalam aplikasi dan apabila data salah maka akan kembali ke form input data pembayaran.

c) Sequence Diagram

Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu, untuk menggambar diagaram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.



Gambar 6 *Sequence Diagram* Halaman Pendaftaran

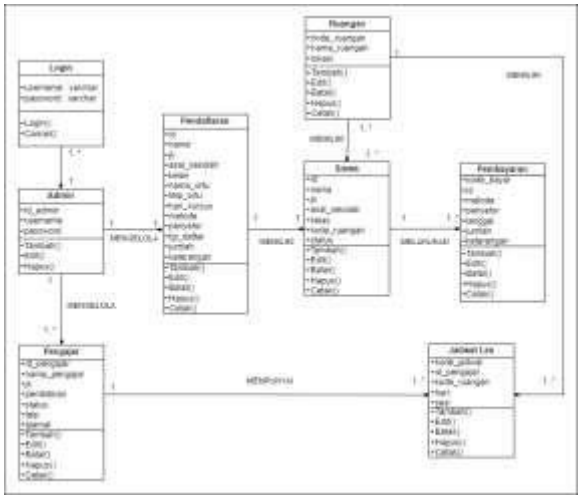


Gambar 7 *Sequence Diagram* Halaman Pembayaran

Berdasarkan Gambar 6 dan 7 dapat dilihat bahwa alur *sequence diagram* mirip dimana pada halaman pendaftaran, tampilan pertama berupa index yang berisi data pendaftaran dan data pembayaran. *Admin* dapat melakukan tambah data, edit data, hapus data, dan mencetak laporan serta mencetak kwitansi.

- Pemodelan Data Sistem

Untuk pemodelan data sistem pada pengembangan aplikasi ini menggunakan UML *Class Diagram*. Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi[6].



Gambar 8 *Class Diagram* yang Diusulkan

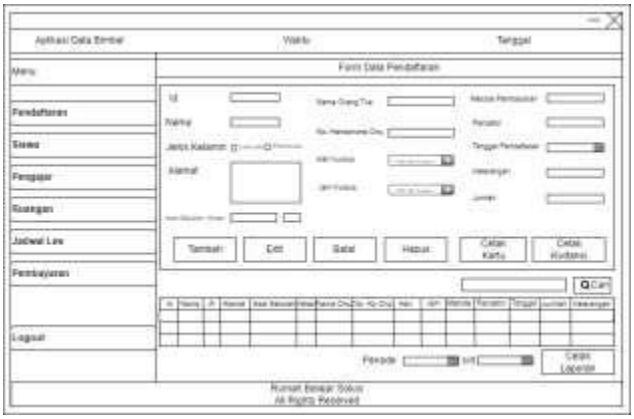
Berdasarkan Gambar 8, *class diagram* memiliki beberapa entitas diantaranya login, admin, siswa, pendaftaran, pembayaran, ruangan, jadwal les, pengajar. Proses login dapat dilakukan oleh satu orang yaitu admin. *Class diagram* hanya memiliki satu level *user* yaitu admin yang bertugas untuk mengelola semua menu yang terdapat pada aplikasi yang diusulkan. Pada *class* diagram ini satu siswa melakukan satu pendaftaran dan juga melakukan banyak pembayaran untuk tiap bulannya. Pengajar mempunyai banyak jadwal les. Ruangan memiliki banyak siswa dan juga banyak jadwal les.

- Perancangan Antarmuka Sistem

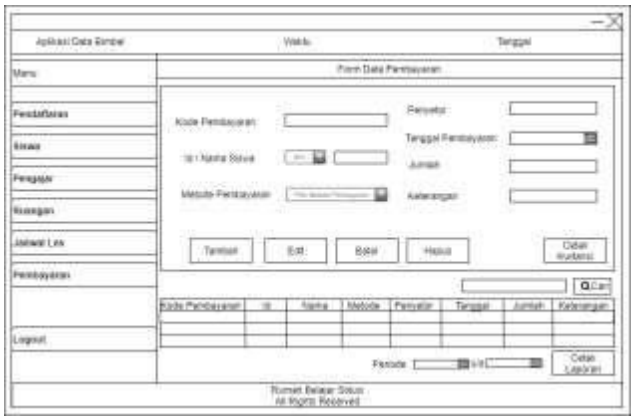
Perancangan antarmuka sistem terdapat dua yaitu perancangan antarmuka masukkan dan perancangan antarmuka keluaran.

a) Perancangan Antarmuka Masukan

Perancangan antarmuka masukkan merupakan proses data yang diinputkan ke sistem. Perancangan antarmuka halaman pendaftaran dan halaman pembayaran dapat dilihat pada Gambar 9 dan Gambar 10.



Gambar 9. Perancangan Antarmuka Halaman Pendaftaran



Gambar 10. Perancangan Antarmuka Halaman Pembayaran

b) Perancangan Antarmuka Keluaran

Perancangan antarmuka keluaran merupakan proses yang dihasilkan dari data-data yang diinputkan. Perancangan antarmuka keluaran terdiri dari kwitansi dan laporandapat dilihat pada gambar di bawah ini :

Gambar 11. Perancangan Antarmuka Kwitansi Pembayaran

Gambar 12. Perancangan Antarmuka Laporan Pembayaran

3. *Coding* (Pengkodean)

   Pada tahap ini dilakukan proses pembuatan kode program pada *software* yang digunakan dalam pengembangan aplikasi. Selain itu dilakukan juga pengujian kode program yang betujuan untuk mengurangi waktu pengembangan dan meminimalisir adanya *bugs* atau *errors*.

   - Pada fase pengkodean ini, aplikasi yang dibuat menggunakan bahasa pemrograman Java dengan *database* MySQL. Dalam penulisan *coding* menggunakan aplikasi NetBeans IDE 8.2 sebagai media untuk menulis, melakukan kompilasi, serta mencari kesalahan pada program.

     Java merupakan pemrograman yang sangat popular karena rentang aplikasi yang bisa dibuat menggunakan bahasa ini sangatlah luas, mulai dari komputer hingga smartphone. Bahasa pemrograman Java dikembangkan pertama kali oleh Sun Microsystemyang dimulai oleh James Gosling dan dirilis pada tahun 1995. Java bersifat *Write Once, Run Anywhere* (program yang ditulis satu kali dan dapat berjalan pada banyak platform)[2].

4. *Testing* (Pengujian)

   Tahap akhir untuk menguji apakah aplikasi yang dibangun sesuai dengan fitur dan fungsionalitas yang diharapkan oleh pengguna. Untuk melakukan pengujian terdapat 2 pengujian yaitu pengujian *black box* dan pengujian *white box*.

   - Pengujian *Black Box*

     Pada pengujian *blackbox* berfokus pada persyaratan fungsional perangkat lunak yang artinya teknik pengujian *blackbox* memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program [5].

Tabel 2 Hasil Uji Halaman Pendaftaran

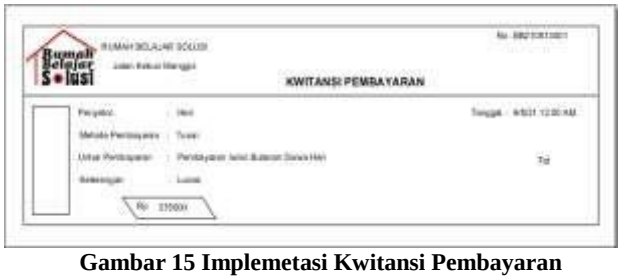
Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Keterangan
Menekan Button Tambah untuk isi data lengkap dan simpan	Sistem Menerima	Data berhasil disimpan di database dan tersimpan pada tabel data pendaftaran	Diterima
Menekan Button Tambah untuk tidak mengisi salah satu data yang ada	Sistem Menolak	Data tidak berhasil disimpan di database dan muncul pemberitahuan	Diterima
Menekan Button Edit untuk isi data lengkap dan simpan	Sistem Menerima	Data berhasil diedit dan tersimpan di database dan tersimpan pada tabel data pendaftaran	Diterima
Menekan Button Hapus	Sistem Menerima	Data yang ingin dihapuskan otomatis langsung terhapus	Diterima
Menekan Button Hapus	Sistem Menerima	Data berhasil terhapus dari database	Diterima
Menekan Button Cari	Sistem Menerima	Menampilkan data yang sesuai dengan pencarian	Diterima
Menekan Button Cetak Kwitansi sesuai data yang dipilih	Sistem Menerima	Menampilkan kwitansi sesuai dengan data yang dipilih	Diterima
Menekan Button Cetak Kartu Siswa sesuai data yang dipilih	Sistem Menerima	Menampilkan kartu siswa sesuai dengan data yang dipilih	Diterima
Menekan Button Cetak Laporan dengan rentang periode	Sistem Menerima	Menampilkan laporan sesuai dengan periode yang ditentukan	Diterima

Tabel 3 Hasil Uji Halaman Pembayaran

Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Keterangan
Menekan Button Tambah untuk isi data lengkap dan simpan	Sistem Menerima	Data berhasil disimpan di database dan tersimpan pada tabel data pembayaran siswa bimbel	Diterima
Menekan Button Tambah untuk tidak mengisi salah satu data yang ada	Sistem Menolak	Data tidak berhasil disimpan di database dan muncul pemberitahuan	Diterima
Menekan Button Edit untuk isi data lengkap dan simpan	Sistem Menerima	Data berhasil diedit dan tersimpan di database dan tersimpan pada tabel data pembayaran siswa bimbel	Diterima
Menekan Button Batal	Sistem Menerima	Data yang ingin dimasukkan otomatis langsung terhapus	Diterima
Menekan Button Hapus	Sistem Menerima	Data berhasil terhapus dari database	Diterima
Menekan Button Cari	Sistem Menerima	Menampilkan data yang sesuai dengan pencarian	Diterima
Menekan Button Cetak Kwitansi sesuai data yang dipilih	Sistem Menerima	Menampilkan kwitansi sesuai dengan data yang dipilih	Diterima
Menekan Button Cetak Kartu Siswa sesuai data yang dipilih	Sistem Menerima	Menampilkan kartu siswa sesuai dengan data yang dipilih	Diterima
Menekan Button Cetak Laporan dengan rentang periode	Sistem Menerima	Menampilkan laporan sesuai dengan periode yang ditentukan	Diterima

Gambar 13 Implementasi Halaman Pendaftaran

Gambar 14 Implementasi Halaman Pembayaran



Gambar 15 Implemetasi Kwitansi Pembayaran

Gambar 16 Implementasi Laporan Pembayaran

- Pengujian *White Box*

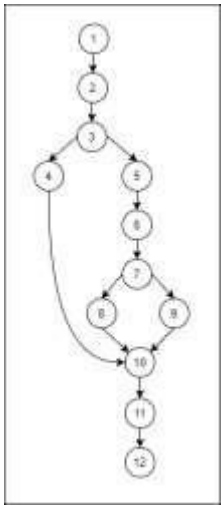
Pengujian *whitebox* disebut juga pengujian kotak kaca (*glass box testing*) merupakan sebuah filosofi perancangan *test case* yang menggunakan struktur kontrol yang dijelaskan sebagai bagian dari perancangan peringkat komponen untuk menghasilkan *test case* [5].

Suatu cyclomatic complexity yang tinggi menunjukkan prosedur yang kompleks, sulit untuk dipahami, diuji dan dipelihara. Ada hubungan antara cyclomatic complexity dan resiko dalam prosedur[1]. Hubungannya ditunjukkan pada tabel 4 dibawah ini:

Tabel 4 Hubungan *Cylomatic Complexity* dan Reskio

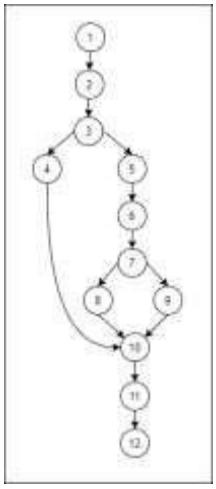
CC	Type of Procedure	Risk
1-4	A simple procedure	Low
5-10	A well structured and stable procedure	Low
11-20	A more complex procedure	Moderate
21-50	A complex procedure, alarming	High
>50	An error-prone, extremely troublesome, untestable procedure	Very High

Gambar 17 *Source Code* Pendaftaran



Gambar 18 Grafik Alir Pendaftaran

Gambar 19 *Source Code* Pembayaran



Gambar 20 Grafik Alir Pembayaran

Berdasarkan grafik alir (*flow graph*) pada gambar di atas, dapat dihitung cyclomatic complexity dari *flow graph* di atas. Berikut adalah perhitungan cyclomatic complexity :

V(G) = E – N + 2

Keterangan : E = jumlah *edge* (anak panah) grafik alir

N = jumlah *node* grafik alir

V(G) = 13 – 12 + 2 = 1 + 2 = 3

Setelah dihitung nilai *cylomatic complexity*, didapatkan bahwa V(G) = 3. Path yang dihasilkan yaitu :

Tabel 5 *Path* yang dihasilkan

Path	**Hasil**	**Tipe prosedur**	**Resiko**
1	1-2-3-4-10-11-12	Prosedur sederhana	Low
2	1-2-3-5-6-7-8-10-11-12	Prosedur kompleks	Moderate
3	1-2-3-5-6-7-9-10-11-12	Prosedur kompleks	Moderate

IV. KESIMPULAN DAN SARAN

A. Kesimpulan

Setelah melakukan kerja praktek di Bimbel Rumah Belajar Solusi dan melihat kinerja yang dilakukan di bimbel tersebut, peneliti mengamati bahwa proses pengelolaan data bimbel masih dilakukan secara manual. Maka dari itu diperlukannya suatu aplikasi untuk membantu proses pengolahan data tersebut. Dalam hal ini, dibuatlah aplikasi pengolahan data bimbel di Rumah Belajar Solusi. Adapun hasil dari pembuatan aplikasi yang telah dibuat adalah sebagai berikut :

1. Aplikasi dapat digunakan untuk mengolah data-data bimbel seperti data pendaftaran, data siswa, data pengajar, data ruangan, data jadwal les, dan data pembayaran.
2. Keamanan data yang terjamin dikarenakan akses yang dibatasin hanya untuk admin yang ditunjuk pemilik bimbel sehingga informasi aman dan terlindungi.
3. Waktu yang diperlukan untuk pengolahan data, pencarian data, dan pembuatan kwitansi serta laporan lebih cepat sehingga menghemat waktu.
4. Meminimalisir terjadinya kesalahan karena sistem telah terkomputerisasi.
5. Penyimpanan data di perangkat lunak *database* sehingga memudahkan pengguna untuk mem*back-up* data sehingga mengurangi risiko terjadinya kehilangan dan kerusakan data.

B. Saran

Adapun saran yang diberikan untuk pengembangan sistem di tahap selanjutnya yaitu :

1. Perlunya dilakukan modifikasi *GUI design* pada tampilan aplikasi agar telihat lebih bagus dan menarik.
2. Perlu adanya sumber daya manusia untuk melakukan *maintenance* secara berkala.
3. Perlunya pengembangan aplikasi agar dapat menghasilkan laporan terkait proses pembelajaran siswa di bimbel Rumah Belajar Solusi.

DAFTAR PUSTAKA

[1] Aviosto. "*Complexity Metrics*". https://www . aivosto . com / project / help /pm-complexity.html. Diakses pada tanggal 27 Januari 2022.

[2] Enterprise, Jubulee.2015. *Mengenal Java dan Database dengan Netbeans*. Jakarta : PT. Elex Media Komputindo.

[3] Gata, Windu, dan Grace Gata. 2013. *Sukses Membangun Aplikasi Penjualan Dengan Java*. Jakarta : Elex Media Komputindo.

[4] Jogiyanto, H. M. 2005. *Analisis dan Desain* (*Sistem Informasi, Pendekatan Terstruktur Teori dan Praktik Aplikasi Bisnis*). Yogyakarta : ANDI.

[5] Pressman, Roger S. 2010. *Rekayasa Perangkat Lunak 1 Edisi 7 (Buku Satu)*. Yogyakarta : ANDI.

[6] Shalahuddin, M. dan Rosa A.S. 2013. *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung : Informatika.

Enjel Lia. Lahir di Palembang tanggal 24 Februari 2001. Saat ini merupakan mahasiswa Universitas Katolik Musi Charitas Palembang dengan program studi S1 Sistem Informasi Angkatan 2018. Merupakan penerima beasiswa Siswa Berprestasi Universitas Katolik Musi Charitas. Saya pernah tergabung dalam Himpunan Mahasiswa Program Studi Sistem Informasi. Riwayat pendidikan saya yaitu di SD Baptis Palembang, SMP Xaverius 1 Palembang, SMA Xaverius 1 Palembang.

Stefanus Setyo Wibagso, S.Kom., M.Kom. lahir di Kota Palembang pada tanggal 14 bulan November tahun 1981. Penulis mendapatkan gelar Sarjana Komputer (S.Kom) pada tahun 2006 dari STMIK Palcomtech Palembang Jurusan Sistem Informasi Bisnis, kemudian melanjutkan pendidikan S2 di Program Pasca Sarjana MTI Universitas Bina Darma Jurusan Software Engineering, Palembang, Indonesia, serta menyelesaikan pendidikan Master pada tahun 2012. Saat ini bekerja sebagai dosen pada Universitas Katolik Musi Charitas Palembang Fakultas Sains & Teknologi pada Program Studi Sistem Informasi. Adapun bidang ilmu yang ditekuni saat ini adalah Web Programming, Java, Joomla Content Management System dan Enterprise Information System, .