

# RANCANG BANGUN SISTEM INFORMASI ADMINISTRASI PENDAFTARAN KURSUS

(STUDI KASUS: *GHIRBANT ENGLISH COURSE-PANDEGLANG*)

Anharudin<sup>1</sup>, Aria Nurdin<sup>2</sup>

Program Studi Sistem Komputer Fakultas Teknologi Informasi Universitas Serang Raya<sup>1</sup>  
Program Studi Teknik Informatika Fakultas Teknologi Informasi Universitas Serang Raya<sup>2</sup>  
[anhar.dean@gmail.com](mailto:anhar.dean@gmail.com)<sup>1</sup>, [arianurdin@gmail.com](mailto:arianurdin@gmail.com)<sup>2</sup>

**Abstrak** – Lembaga Bahasa Internasional Ghibrant *English Course* (LBI-GEC) salah satu lembaga pendidikan non formal atau kursus. Ghibrant *English Course* beralamat di Jl. Yusuf Martadilaga, Kabayan Cibunut Pandeglang. Yang mengatur segala hal yang berkaitan dengan penyelenggaraan pendidikan maupun yang lainnya, salah satunya dalam hal sistem administrasi pendaftaran kursus yang masih menggunakan cara konvensional. Aplikasi sistem informasi administrasi pendaftaran ghibrant dirancang dengan pemodelan *Unified Modeling Language* (UML) yang terdiri dari *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, *Class Diagram*, dibuat dengan menggunakan Borland Delphi 7 dan *database MySQL* yang dapat digunakan di Windows 7, windows 8, windows 10. Sehingga membuat proses administrasi pendaftaran di ghibrant pandeglang menjadi lebih praktis dan optimal. Dengan adanya aplikasi sistem informasi ghibrant *English course* ini dapat mempermudah pengolahan data administrasi pendaftaran. Sehingga dapat menghasilkan informasi yang cukup akurat dan tidak banyak melakukan kesalahan dalam pengolahan data administrasi.

**Kata Kunci:** Pendaftaram, Sistem Informasi, *Unified Modeling Language*

## I. PENDAHULUAN

Pada era globalisasi saat ini perkembangan teknologi semakin pesat, kebutuhan informasi yang cepat, tepat dan akurat ini sangat penting. Berbagai lembaga pada bidang instansi pemerintahan dan pendidikan berusaha mengembangkan kualitas dengan melakukan perubahan memanfaatkan teknologi informasi. Informasi sudah semakin mudah diperoleh, bervariasi bentuknya, dan sudah semakin banyak kegunaannya. Sistem informasi sangat diperlukan oleh berbagai pihak untuk mengelola data menjadi informasi sehingga pengambilan keputusan informasi tersebut dapat menghasilkan keputusan yang baik.

Lembaga Bahasa Internasional Ghibrant *English Course* (LBI-GEC) salah satu lembaga pendidikan non formal atau kursus. Ghibrant *English Course* beralamat di Jl. Yusuf Martadilaga, Kabayan Cibunut Pandeglang. Yang mengatur segala hal yang berkaitan dengan penyelenggaraan pendidikan maupun yang lainnya, salah satunya dalam hal sistem administrasi pendaftaran kursus yang masih menggunakan cara konvensional.

Adapun kondisi yang terjadi pada LBI GEC yang banyak diminati di kabupaten Pandeglang dengan bertambahnya jumlah siswa dari tahun ke tahun, khususnya pada bagian administrasi pendaftaran selama ini penyajian dan pengolahan datanya masih bersifat manual yaitu dengan mencatat data-data ke dalam pembukuan. Seperti penyimpanan data siswa sehingga untuk memperoleh informasi yang dibutuhkan terkadang harus mencarinya satu persatu, dan proses

pencarian data juga membutuhkan waktu yang lama sehingga banyak waktu yang terbuang.

Oleh karena itu untuk mendukung peningkatan kualitas yang baik serta mempermudah dan menghemat waktu di Lembaga Bahasa Internasional Ghibrant *English Course* (LBI-GEC) agar lebih baik lagi diperlukannya aplikasi Rancang Bangun Sistem Informasi Administrasi Pendaftaran Ghibrant *English Course* Pandeglang.

## II. KAJIAN PUSTAKA

Krisanti Vonny Anggraeny, Wellia Shinta Sari (2015) Melakukan Penelitian Tentang “Sistem Informasi Administrasi Pada Bimbingan Tes Alumni 70 Jakarta”. pada penelitian ini mengenai administrasi pada bimbingan Tes Alumni 70 Jakarta sistem yang berjalan masih belum efektif dan selalu menginginkan kemudahan ataupun kecepatan informasi yang relevan maka Peneliti merancang sebuah aplikasi sistem informasi administrasi supaya agar mempermudah melakukan pendataan informasi tentang data administrasi dengan mudah, cepat dan akurat serta dengan resiko kesalahan yang sangat kecil dan memberikan penyimpanan informasi dalam jumlah besar.

Elly Meilani, Heny Sumarti, Mardiani dan Della Oktaviany (2014) Melakukan Penelitian Tentang “Sistem Informasi Administrasi Dan Pembelajaran Siswa Berbasis Desktop Dan Website Pada Family Course” Peneliti merancang suatu pengembangan sistem administrasi yang dapat membantu dalam pengolahan data administrasi bimbingan

secara terorganisir dan juga mengembangkan sistem pembelajaran secara *online* maka dibuat aplikasi berbasis *desktop* sebagai pengelola data administrasi seperti data siswa, data pengajar, jadwal dan data pembayaran agar dapat terorganisir dengan baik dan dapat menyajikan kebutuhan akan informasi secara cepat dan tepat.

**Ajeng Melissa, (2013) Melakukan Penelitian Tentang “Sistem Informasi Administrasi Pendidikan Pada Bimbingan Belajar Super Student Club (SSC) Ngaliyan Semarang”.** Peneliti merancang sebuah aplikasi sistem informasi administrasi, Karena belum adanya sistem komputerisasi yang berjalan maka sering terjadi kesalahan dalam penyampaian informasinya. Sistem informasi ini bertujuan untuk membantu meminimalisir kesalahan dalam menyampaikan informasi, sehingga informasi di dapat lebih cepat pengolahan administrasi sehingga lebih menghemat waktu. Perancangan sistem informasi ini juga dapat meningkatkan efektifitas dan efisiensi kerja sehingga kinerja para staf tidak terhambat.

**Deci Irmayani, (2014) Melakukan Penelitian Tentang “Sistem Informasi Administrasi Kesiswaan Pada SMK Pemda Rantauprapat Berbasis Client Server”.** Peneliti merancang sebuah aplikasi untuk mempermudah dalam penyajian sistem pendataan guru dan siswa SMK Swasta Pemda Rantauprapat berbasis *client server* yang terdiri dari sistem pendataan guru, siswa, absensi siswa, absensi guru, mata pelajaran, dan nilai siswa dengan adanya pengolahan administrasi kesiswaan yang baik dapat membantu meningkatkan efisiensi kegiatan belajar dan untuk mencapai tujuan lebih produktif, efektif, efisien dan relevan di bidangnya.

### Pengertian Sistem

Suatu sistem dapat didefinisikan sebagai suatu kesatuan yang terdiri dari dua atau lebih komponen atau subsistem yang berinteraksi untuk mencapai suatu tujuan. Suatu sistem dapat terdiri dari sistem-sistem bagian (subsistem). Misalnya sistem komputer yang terdiri dari subsistem perangkat keras dan subsistem perangkat lunak. Subsistem-subsistem saling berinteraksi dan saling berhubungan membentuk satu kesatuan sehingga tujuan atau sasaran sistem tersebut dapat tercapai.

Menurut Jogiyanto HM (2008: 1), “Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu”.

Menurut Tata Sutabri (2005), “Sistem sebagai suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu”.

### Karakteristik Sistem

Sistem merupakan kumpulan komponen-komponen yang saling berinteraksi satu dengan lainnya membentuk satu kesatuan, sehingga sistem mempunyai

karakteristik menurut (Jogiyanto HM, 2008) sebagai berikut:

1. **Komponen Sistem**, Setiap subsistem atau bagian dari sistem, dimana setiap subsistem mempunyai sifat dari sistem untuk menjalankan suatu fungsi tertentu dan akan mempengaruhi sistem secara keseluruhan, meliputi:
  - a. Tujuan, sasaran rancangan sistem yang hendak dicapai haruslah memuaskan pemakai.
  - b. Variabel, semua unsur yang menyusun sistem, bagi analisis atau perancangan sistem harus dapat menguraikan sistem atau variabel-variabel secara terinci untuk mencari kekuatan, kelemahan, peluang dan ancaman.
  - c. Hubungan, hubungan antar variabel akan terlihat dalam proses.
  - d. Output, kemampuan sistem menghasilkan sesuatu, misalnya informasi daya, suara dan lain-lain
2. **Batasan Sistem**, Batasan yang membatasi antara satu sistem dengan sistem yang lainnya atau antara subsistem dengan subsistem lainnya.
3. **Lingkaran Luar**, Segala sesuatu yang berada di luar sistem yang dapat mempengaruhi sistem.
4. **Penghubung Sistem**, Media yang menjembatani antara suatu sistem dengan sistem yang lainnya atau antara subsistem dengan subsistem lainnya.
5. **Masukan Sistem**, Apa saja yang diterima oleh sistem dari lingkungan luar untuk mencapai tujuan.
6. **Keluaran Sistem**, Apa saja yang dihasilkan oleh sistem dari keluaran untuk mencapai tujuan.
7. **Pengolahan Sistem**, merubah masukan menjadi keluaran.

### Pengertian Informasi

Definisi dari informasi adalah data yang diolah menjadi bentuk yang lebih berarti bagi yang menerimanya. Menurut Raymond Mcleod informasi adalah data yang telah diolah menjadi bentuk yang memiliki arti bagi si penerima dan bermanfaat bagi pengambilan keputusan saat ini atau mendatang. Secara umum informasi dapat didefinisikan sebagai hasil pengolahan data dalam bentuk yang lebih berguna dan lebih berarti bagi penerimannya.

Sumber dari informasi adalah data. Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Kejadian-kejadian adalah sesuatu yang terjadi pada saat tertentu.

Data merupakan bentuk yang masih mentah, belum dapat bercerita banyak sehingga perlu diolah lebih lanjut. Data diolah melalui suatu metode untuk menghasilkan informasi

Data yang diolah melalui suatu model menjadi informasi, penerima kemudian menerima informasi

tersebut, membuat suatu keputusan dan melakukan tindakan, yang berarti menghasilkan suatu tindakan yang lain yang akan membuat sejumlah data kembali.

### Pengertian Sistem Informasi

Pengertian Sistem Informasi adalah suatu sistem yang menyediakan informasi untuk manajemen pengambilan keputusan/kebijakan dan menjalankan operasional dari kombinasi orang-orang, teknologi informasi dan prosedur-prosedur yang terorganisasi atau sistem informasi diartikan sebagai kombinasi dari teknologi informasi dan aktivitas orang yang menggunakan teknologi untuk mendukung operasi dan manajemen. Sedangkan dalam arti luas, sistem informasi diartikan sebagai sistem informasi yang sering digunakan menurut kepada interaksi antara orang, proses, algoritmik, data dan teknologi.

### Borlan Delphi 7

Bahasa pemrograman Delphi adalah pemrograman berorientasi objek yang merupakan pengembangan dari bahasa pemrograman Pascal. Pascal adalah pemrograman berbasis DOS yang dibuat pada tahun 1971 oleh Niklaus Wirth dari Swiss. Kata Pascal diambil dari nama matematikawan dan ahli filsafat dari Perancis, yaitu Blaise Pascal yang lahir tahun 1623 dan meninggal tahun 1662. Pascal dirilis tahun 1983 oleh Borland International, salah satu perusahaan *software* terkemuka di California.

Pada tahun 1993, Borland International mengembangkan bahasa pemrograman Pascal yang bersifat visual yang disebut Delphi dan resmi dipasarkan pada tahun 1995. Pemrograman ini dibuat secara modern yang berjalan di Sistem Operasi Windows mulai dari versinya yang pertama yaitu Delphi 1 dan di tahun-tahun berikutnya Delphi terus dikembangkan mengikuti kebutuhan zaman (Kani, Firmansyah, & Sufandi, 2010).

### Database

Basis data (*database*) adalah suatu kumpulan data yang disusun dalam bentuk tabel-tabel yang saling berkaitan maupun berdiri sendiri dan disimpan secara bersama-sama pada suatu media. Basis data dapat digunakan oleh satu atau lebih program aplikasi secara optimal, data disimpan tanpa mengalami ketergantungan pada program yang akan menggunakannya.

Terdapat beberapa aturan yang harus dipatuhi pada *file* basis data agar dapat memenuhi kriteria sebagai suatu basis data, yaitu:

1. Kerangkapan data, yaitu munculnya data-data yang sama secara berulang-ulang pada *file* basis data.
2. Inkonsistensi data, yaitu munculnya data yang tidak konsisten pada *field* yang sama untuk beberapa *file* dengan kunci yang sama,
3. Data terisolasi, disebabkan oleh pemakaian beberapa *file* basis data. Program aplikasi tidak dapat mengakses *file* tertentu dalam sistem basis data tersebut, kecuali program aplikasi dirubah atau ditambah sehingga seolah-olah ada *file* yang terpisah atau terisolasi terhadap *file* yang lain,
4. Keamanan data, berhubungan dengan masalah keamanan data dalam sistem basis data. Pada

prinsipnya *file* basis data hanya boleh digunakan oleh pemakai tertentu yang mempunyai wewenang untuk mengakses,

5. Integrasi data, berhubungan dengan unjuk kerja sistem agar dapat melakukan kendali atau kontrol pada semua bagian sistem sehingga sistem selalu beroperasi dalam pengendalian penuh.

### MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: *database management system*) atau DBMS yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis di bawah lisensi GNU *General Public License* (GPL), tetapi mereka juga menjual di bawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL *Relational Database Management System* (RDBMS).

MySQL adalah *Relational Database Management System* (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam database sejak lama, yaitu SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian *database*, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. Keandalan suatu sistem *database* (DBMS) dapat diketahui dari cara kerja *optimizer*-nya dalam melakukan proses perintah-perintah SQL, yang dibuat oleh *user* maupun program-program aplikasinya. Sebagai *database server*, MySQL dapat dikatakan lebih unggul dibandingkan *database server* lainnya dalam *query* data. Hal ini terbukti untuk *query* yang dilakukan oleh *single user*, kecepatan *query* MySQL bisa sepuluh kali lebih cepat dari PostgreSQL dan lima kali lebih cepat dibandingkan *Interbase*.

### Unified Modeling Language (UML)

UML ialah terdiri dari kata *Unified Modelling Language* yang merupakan suatu alat untuk seorang design sistem atau seorang *analyst* untuk membuat pemodelan rancangan yang akan nantinya diserahkan kepada *programmer* dan diterjemahkan ke dalam bahasa pemrograman apapun yang bisa menghasilkan program baru lalu bisa membuat suatu sistem pada perusahaan atau instalasi yang sesuai dengan keinginan pemesan sistem. UML merupakan sebuah standar untuk merancang sebuah model sistem, untuk mengembangkan UML maka harus di perhatikan diagram seperti apa yang dibutuhkan dalam perancangan untuk membuat model sistem yaitu sebagai berikut:

#### a. Use Case

Diagram *Use Case* merupakan suatu pemodelan sistem yang terdiri dari aktor lalu dihubungkan dengan *use case* pada sistem yang dibuat, diagram ini menggambarkan bagaimana keterhubungan antara aktor dengan *use case*, aktor disini tidak selalu berupa

manusia sebagai pengelola ataupun *customer*, bisa juga berupa peralatan ataupun suatu sistem lain yang berhubungan dengan sistem yang dikerjakan saat ini.





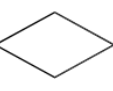
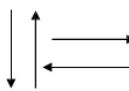
**Tabel 1. Use Case Diagram**

NO	GAMBAR	NAMA	KETERANGAN
1		Actor	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan use case.
2		Include	Menspesifikasikan bahwa use case sumber secara eksplisit.
3		Extend	Menspesifikasikan bahwa use case target memperluas perilaku dari use case sumber pada suatu titik yang diberikan.
4		Association	Apa yang menghubungkan antara objek satu dengan objek lainnya.
5		System	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
6		Use Case	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor

**b. Diagram Aktivitas**

*Diagram Activity* bisa juga di sebut *flowmap* ini merupakan pemodelan berupa arus *flow* yang di mulai dari tanda *start* sampai *end* ciri dari *flowmap* itu berupa suatu kumpulan entitas yang melakukan proses sistem sebagaimana mestinya yang saling terhubung.

**Tabel 2. Activity Diagram**

NO	GAMBAR	NAMA	KETERANGAN
1		activity	Memperlihatkan bagaimana masing-masing kelas antar muka saling berinteraksi satu sama lain
2		action	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		Initial node	Bagaimana objek dibentuk atau diawali
4		Activity final node	Bagaimana objek dibentuk atau diakhiri
5		decision	Digunakan untuk menggambarkan suatu keputusan / tindakan yang harus diambil pada kondisi tertentu
6		Line connector	Digunakan untuk menghubungkan satu symbol dengan symbol lainnya

**C. Diagram Sequensial**

Diagram ini menggambarkan suatu aliran fungsi dari *use case*, jadi intinya suatu aliran yang berfungsi sebagai pemodelan yang dihubungkan dengan *use case* dan fungsi nya masing masing, agar bisa dilihat fungsi satu *use case* dengan *use case* lainnya.







**Tabel 3. Sequensial Diagram**

No.	Gambar	Nama	Keterangan
1		LifeLine	objek <i>entity</i> , antarmuka yang saling berinteraksi.
2		Message	spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
3		Message	spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi

**d. Diagram Kelas**

Diagram kelas (*Class Diagram*) itu merupakan penggambaran dari sistem itu sendiri yang berupa program ataupun suatu menu form pilihan yang berisi tentang apa-apa saja yang saling berdekatan digambarkan dengan *class diagram* berupa tabel kotak-kotak kecil berisi suatu kata kunci dan isinya masing-masing yang berhubungan, dalam merancang *class diagram* ini harus membutuhkan ketelitian dan pemikiran yang dapat diimplementasikan ke dalam program sistem yang akan dibuat nanti.

**Tabel 4. Class Diagram**

NO	GAMBAR	NAMA	KETERANGAN
1		Generalization	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).
2		Class	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
3		Collaboration	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
4		Realization	Operasi yang benar-benar dilakukan oleh suatu objek.
5		Dependency	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
6		Association	Apa yang menghubungkan antara objek satu dengan objek lainnya

Dalam penelitian ini dibagi menjadi beberapa tahapan metode penelitian yaitu:

**A. Metode Pengumpulan Data**

**1. Wawancara**

Teknik pengumpulan data dengan cara tanya jawab secara langsung di lapangan yaitu dengan melihat

langsung cara pembelajaran antara guru dan siswa di dalam kelas.

## 2. Observasi

Teknik pengumpulan data dengan mengadakan penelitian dan peninjauan secara langsung terhadap permasalahan yang diambil.

## 3. Kajian Pustaka

Teknik pengumpulan data dengan cara mengumpulkan informasi dan data melalui bantuan teks-teks tertulis maupun *soft copy* seperti jurnal, paper dokumen dan bacaan-bacaan yang berkaitan dengan judul penelitian.

## B. Prototyping

Menurut Raymond McLeod, *prototype* didefinisikan sebagai alat yang memberikan ide bagi pembuat maupun pemakai potensial tentang cara sistem berfungsi dalam bentuk lengkapnya, dan proses untuk menghasilkan sebuah *prototype* disebut *prototyping*.

*Prototyping* adalah proses pembuatan model sederhana *software* yang memungkinkan pengguna memiliki gambaran dasar tentang program serta melakukan pengujian awal. *Prototyping* memberikan fasilitas bagi pengembang dan pemakai untuk saling berinteraksi selama proses pembuatan, sehingga pengembang dapat dengan mudah memodelkan perangkat lunak yang akan dibuat. *Prototyping* merupakan salah satu metode pengembangan perangkat lunak yang banyak digunakan.

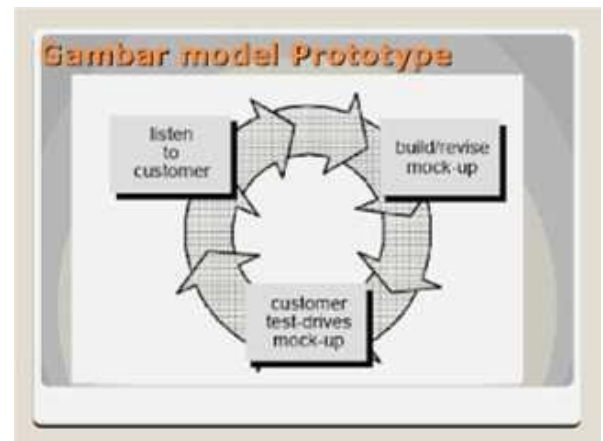
Model tersebut dapat berupa tiga bentuk:

1. Prototipe kertas atau model berbasis komputer yang menjelaskan bagaimana interaksi antara pemakai dan komputer.
2. Prototipe yang mengimplementasikan beberapa bagian fungsi dari perangkat lunak yang sesungguhnya. Dengan cara ini pemakai akan lebih mendapatkan gambaran tentang program yang akan dihasilkan, sehingga dapat menjabarkan lebih rinci kebutuhannya.
3. Menggunakan perangkat lunak yang sudah ada. Seringkali pembuat *software* memiliki beberapa program yang sebagian dari program tersebut mirip dengan program yang akan dibuat.

Di dalam proses pengembangan, sering kali pemakai / pelanggan hanya dapat mendefinisikan tujuan dan penggunaan *software* yang dibutuhkan, tetapi tidak dapat mendefinisikan secara rinci kebutuhan masukan, pengolahan, dan keluarannya. Di sisi lain, pembuat *software* tidak memiliki kepastian akan hal tersebut. Hal ini menyebabkan pengembang kurang memperhatikan efisiensi algoritma, kemampuan sistem operasi dan *interface* yang menghubungkan manusia dan komputer. Untuk menyelaraskan antara pelanggan dan pengembang, maka harus dibutuhkan kerjasama yang baik di antara keduanya sehingga pengembang akan mengetahui dengan benar apa yang diinginkan pelanggan dengan tidak mengesampingkan segi-segi teknis. Dan pelanggan akan mengetahui proses-proses dalam menyelesaikan sistem yang diinginkan. Dengan demikian akan menghasilkan sistem sesuai dengan jadwal waktu penyelesaian yang telah ditentukan.

Kunci agar model *prototype* ini berhasil dengan baik adalah dengan mendefinisikan aturan-aturan main pada saat awal, yaitu pelanggan dan pengembang harus setuju bahwa *prototype* dibangun untuk mendefinisikan kebutuhan.

*Prototyping* merupakan Javascript Framework yang dibuat untuk lebih memudahkan proses dalam membangun aplikasi berbasis *web*. Metode *prototyping* sebagai suatu paradigma baru dalam pengembangan sistem informasi, tidak hanya sekedar suatu evolusi dari metode pengembangan sistem informasi yang sudah ada, tetapi sekaligus merupakan revolusi dalam pengembangan sistem informasi manajemen.



**Gambar 1. Model Prototyping**

Proses-proses tersebut dapat dijelaskan sebagai berikut:

1. Pengumpulan kebutuhan: *developer* dan klien bertemu dan menentukan tujuan umum, kebutuhan yang diketahui dan gambaran bagian-bagian yang akan dibutuhkan berikutnya;
2. Perancangan: perancangan dilakukan cepat dan rancangan mewakili semua aspek *software* yang diketahui, dan rancangan ini menjadi dasar pembuatan *prototype*;
3. Evaluasi *Prototype*: klien mengevaluasi *prototype* yang dibuat dan digunakan untuk memperjelas kebutuhan *software*.

Pendekatan utama *prototyping*, yaitu: EVOLUTIONARY. Pada metode ini, prototipenya tidak dibuang tetapi digunakan untuk iterasi desain berikutnya. Dalam hal ini, sistem atau produk yang sebenarnya dipandang sebagai evolusi dari versi awal yang sangat terbatas menuju produk final atau produk akhir.

*Prototyping* bekerja dengan baik pada penerapan-penerapan yang berciri sebagai berikut:

1. Resiko tinggi yaitu untuk masalah-masalah yang tidak terstruktur dengan baik, ada perubahan yang besar dari waktu ke waktu, dan adanya persyaratan data yang tidak menentu.
2. Interaksi pemakai penting. Sehingga sistem harus menyediakan dialog *online* antara pelanggan dan komputer.
3. Waktu penyelesaian yang cepat.



4. Perilaku pemakai yang sulit ditebak.
5. Sistem yang inovatif yaitu system yang membutuhkan cara penyelesaian masalah dan penggunaan perangkat keras yang mutakhir. Untuk memodelkan sebuah perangkat lunak, metode *prototyping* memiliki tahapan-tahapan di dalam proses pengembangannya. Tahapan inilah yang menentukan keberhasilan dari sebuah *software*. Pengembang perangkat lunak harus memperhatikan tahapan dalam metode *prototyping* agar *software* akhirnya dapat diterima oleh pemakai.

Dan tahapan-tahapan dalam *prototyping* tersebut adalah sebagai berikut:

1. Pengumpulan kebutuhan  
Pelanggan dan pengembang bersama-sama mendefinisikan format seluruh perangkat lunak, mengidentifikasi semua kebutuhan, dan garis besar sistem yang akan dibuat.
2. Membangun *prototyping*  
Membangun *prototyping* dengan membuat perancangan sementara yang berfokus pada penyajian kepada pelanggan (misalnya dengan membuat input dan format output).
3. Evaluasi *prototyping*  
Evaluasi ini dilakukan oleh pelanggan apakah *prototyping* yang sudah dibangun sudah sesuai dengan keinginan pelanggan. Jika sudah sesuai maka langkah keempat akan diambil. Jika tidak, maka *prototyping* direvisi dengan mengulang langkah 1, 2, dan 3.
4. Mengkodekan sistem  
Dalam tahap ini *prototyping* yang sudah disepakati diterjemahkan ke dalam bahasa pemrograman yang sesuai.
5. Menguji sistem  
Setelah sistem sudah menjadi suatu perangkat lunak yang siap pakai, harus dites dahulu sebelum digunakan. Pengujian ini dilakukan dengan *Black Box*.
6. Evaluasi Sistem  
Pelanggan mengevaluasi apakah sistem yang sudah jadi sudah sesuai dengan yang diharapkan. Jika sudah, maka langkah ketujuh dilakukan, jika belum maka mengulangi langkah 4 dan 5.
7. Menggunakan sistem  
Perangkat lunak yang telah diuji dan diterima pelanggan siap untuk digunakan.

Berikut ini adalah jenis-jenis *prototyping*, yaitu:

1. *Feasibility prototyping*  
Digunakan untuk menguji kelayakan dari teknologi yang akan digunakan untuk sistem informasi yang akan disusun.
2. *Requirement prototyping*  
Digunakan untuk mengetahui kebutuhan aktivitas bisnis *user*.
3. *Desain prototyping*  
Digunakan untuk mendorong perancangan sistem informasi yang akan digunakan.
4. *Implementation prototyping*

Merupakan lanjutan dari rancangan, *prototype* ini langsung disusun sebagai suatu sistem informasi yang akan digunakan.

Teknik-teknik *prototyping* meliputi:

1. Perancangan Model  
Perancangan awal *software* oleh pengembang untuk dimodelkan sebagai gambaran awal kepada pengguna.
2. Perancangan Dialog  
Perancangan menu-menu pada *software* yang dibuat, dengan maksud agar pengguna dapat dengan mudah menggunakannya.
3. Simulasi  
Proses percobaan *software* kepada calon *user* sebelum *software* dinyatakan layak pakai.

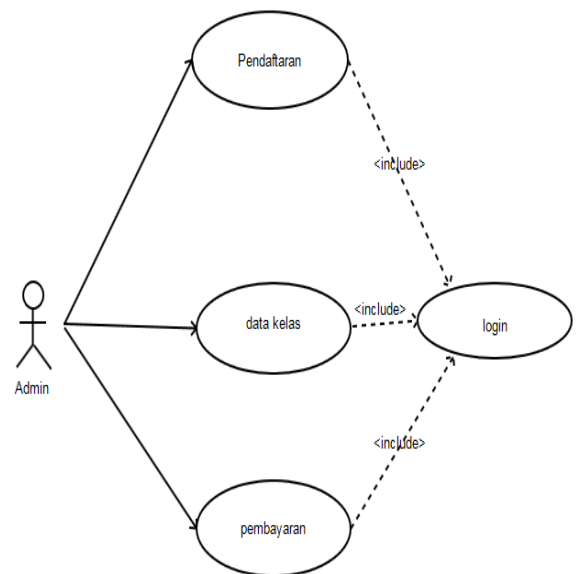
#### IV HASIL DAN PEMBAHASAN

##### Perancangan Sistem

Perancangan sistem dibuat untuk menggambarkan alur proses data yang terjadi dalam suatu sistem informasi. Proses perancangan sistem lebih jelasnya akan dijabarkan pada UML (*Unified Modeling Language*) dengan dideskripsikan menggunakan *Usecase Diagram*, *Class Diagram*, *Sequence Diagram* dan *Activity Diagram*.

##### Use Case Diagram

*Use case diagram* menjelaskan mengenai siapa saja yang berhubungan langsung dengan sistem yang ada. Pada sistem pengelola aplikasi ini terdapat 1 aktor yang berhubungan dengan aplikasi tersebut diantaranya admin, aksi-aksi yang dilakukan oleh aktor tersebut dapat dilihat pada gambar *use case diagram* berikut ini *Use Case Diagram*.



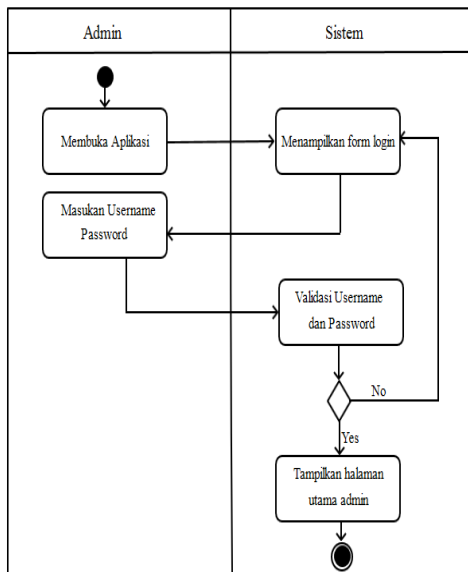
Gambar 2. Use Case Diagram Pendaftaran

**Activity Diagram**

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal decision yang mungkin terjadi dan bagaimana mereka berakhir.

**1. Activity Diagram Login**

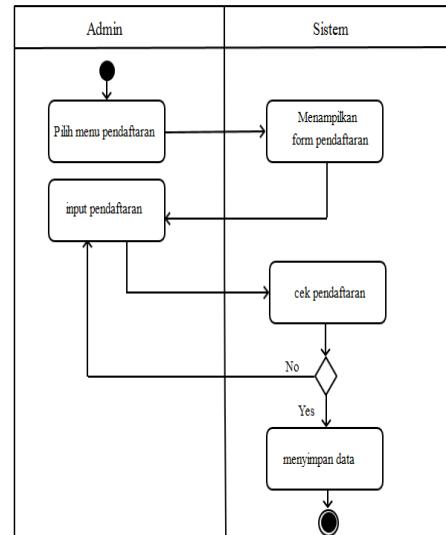
*Activity Diagram Login* ini menjelaskan tentang bagaimana Admin/Staff memiliki otoritas lebih untuk mengakses penuh aplikasi. *activity diagram login* ini admin memasukkan *username* dan *password*. Kemudian sistem melakukan validasi *password* yang diinput oleh admin/staff, apabila terjadi kesalahan maka sistem tidak dapat meneruskan ke *form* yang diinginkan dan admin harus *login* kembali untuk dapat mengakses *form* yang diinginkan.



**Gambar 3. Activity Diagram Login**

**Activity Diagram Pendaftaran**

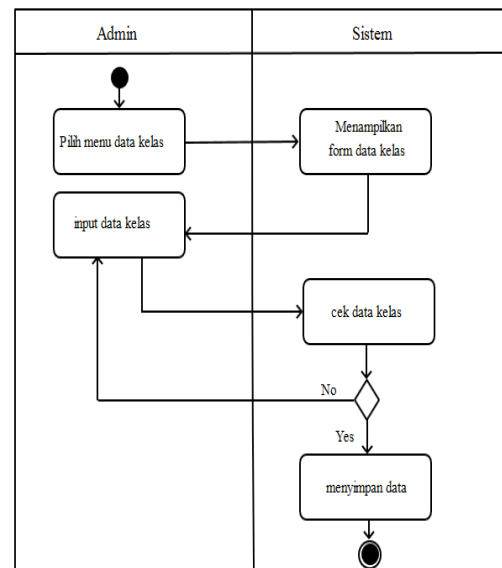
*Activity diagram pendaftaran* menjelaskan tentang proses admin/staff menginput pendaftaran, di mulai dari membuka aplikasi hingga proses input pendaftaran selesai, prosesnya dapat dilihat pada gambar berikut.



**Gambar 4. Activity Diagram Input Pendaftaran**

**Activity Diagram Data Kelas**

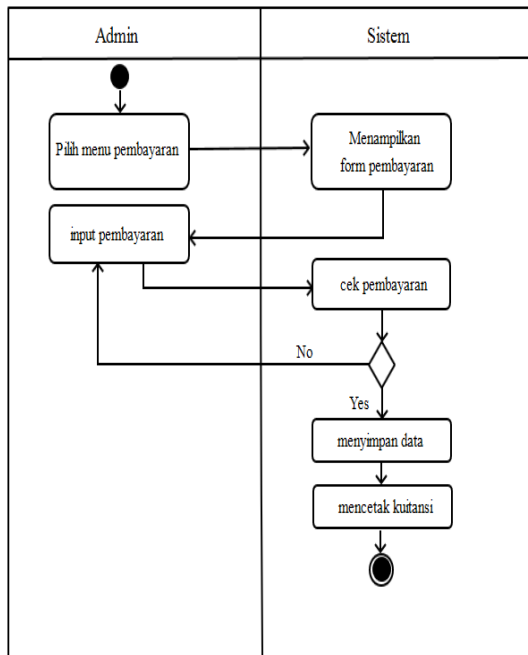
*Activity diagram data kelas* menjelaskan proses tentang input data kelas, dimulai dari membuka aplikasi hingga proses input data kelas selesai, prosesnya dapat dilihat pada gambar berikut.



**Gambar 5. Activity Diagram Input Data Kelas**

**Activity Diagram Data Pembayaran**

*Activity diagram data pembayaran* menjelaskan proses admin/staff tentang meinput data pembayaran, dimulai dari membuka aplikasi hingga proses input data pembayaran selesai, prosesnya dapat dilihat pada gambar berikut.

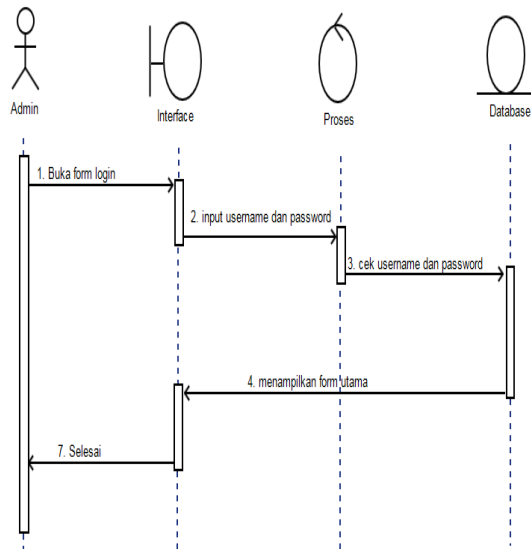


Gambar 6. Activity Diagram Input Pembayaran

**Sequence Diagram**

1. *Sequence Diagram Login*

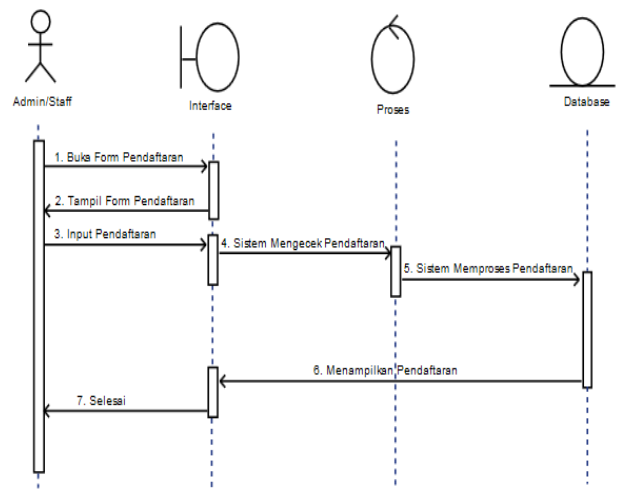
Berikut ini adalah *sequence diagram menu login*.



Gambar 7. Sequence Diagram Login

2. *Sequence Diagram Pendaftaran*

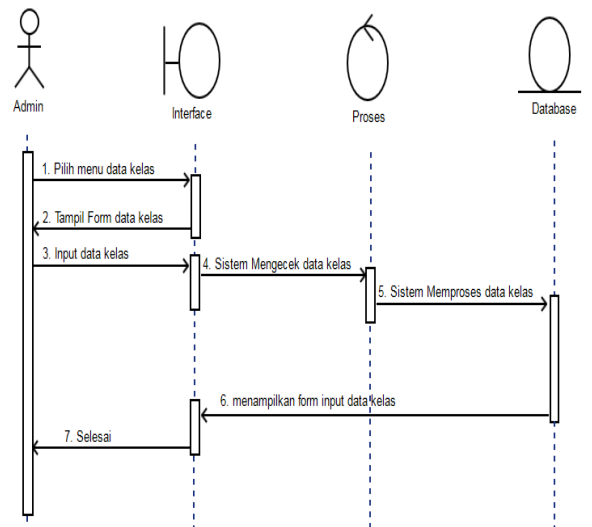
Berikut ini adalah *sequence diagram menu pendaftaran*.



Gambar 8. Sequence Diagram Pendaftaran

3. *Sequence Diagram Data Kelas*

Berikut ini adalah *sequence diagram menu data kelas*.

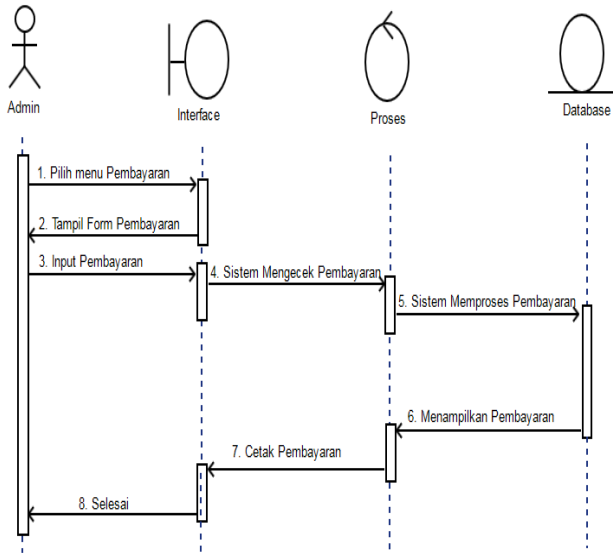


Gambar 9. Sequence Diagram Data Kelas

4. *Sequence Diagram Pembayaran*

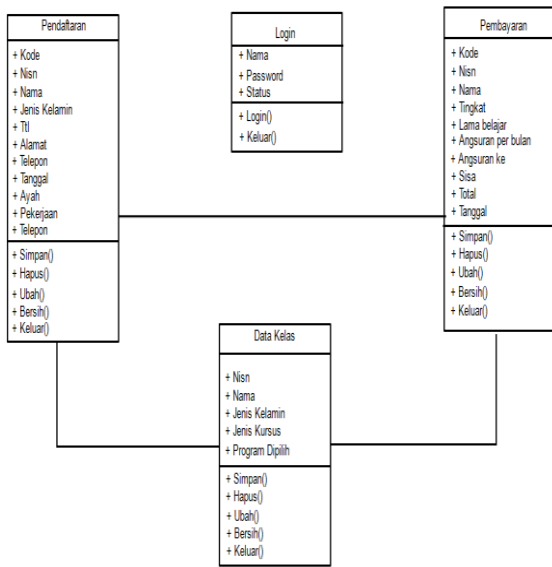
Berikut ini adalah *sequence diagram menu pembayaran*.





Gambar 10. Sequence Diagram Pembayaran

Class Diagram

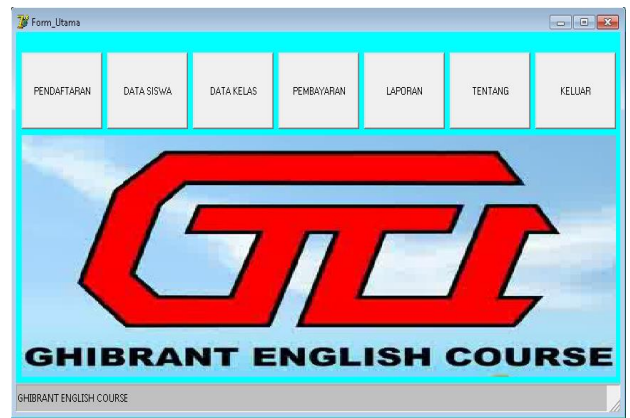


Gambar 11. Class Diagram

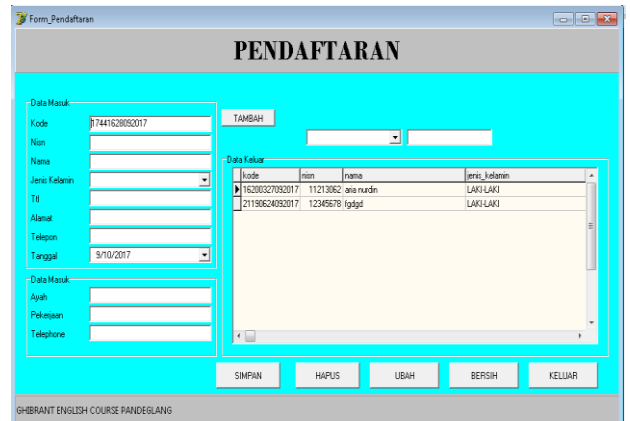
User Interface



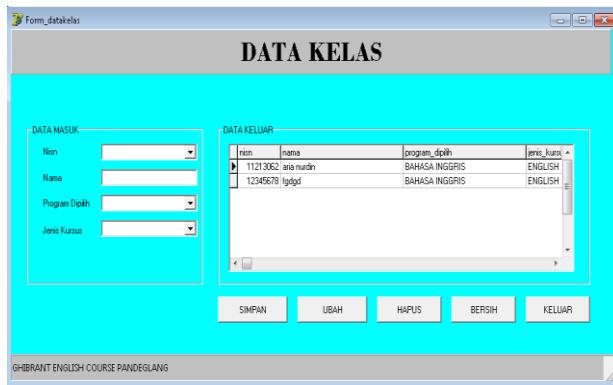
Gambar 12. Menu Login



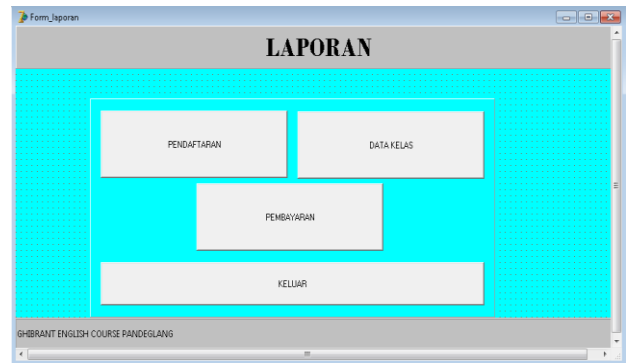
Gambar 13. Halaman Utama



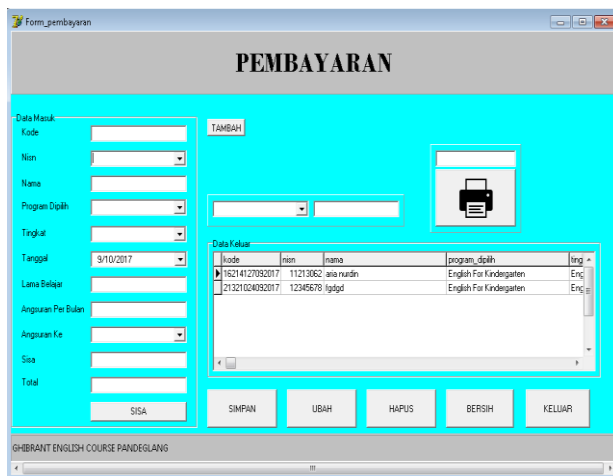
Gambar 13. Halaman Pendaftaran



Gambar 14. Halaman Data Kelas



Gambar 17. Halaman Laporan



Gambar 15. Halaman Pembayaran



Gambar 16. Kuitansi Pembayaran

**Pengujian Sistem**

Pengujian sistem dilakukan untuk menemukan *error* dan kesalahan pada aplikasi, serta untuk menilai aplikasi yang dirancang telah sesuai dengan apa yang diharapkan. Metode pengujian yang digunakan adalah pengujian *Black Box* yaitu pengujian yang didasarkan pada detail aplikasi, fungsi-fungsi yang ada pada aplikasi, dan kesesuaian alur fungsi dengan proses yang diinginkan oleh pengguna.

Tabel 5. Pengujian *Black Box*

Item Pengujian	Detail Pengujian	Hasil	Ket	
			Bisa	Tidak
Halaman Login	Button Masuk	Berjalan dengan Baik	✓	
	Button Keluar	Berjalan dengan Baik	✓	
Halaman Utama	Button Data Siswa	Berjalan dengan Baik	✓	
	Button Data Siswa	Berjalan dengan Baik	✓	
	Button Data Kelas	Berjalan dengan Baik	✓	
	Button Pembayaran	Berjalan dengan Baik	✓	
	Button Laporan	Berjalan dengan Baik	✓	
	Button Tentang	Berjalan dengan Baik	✓	
	Button Keluar	Berjalan dengan Baik	✓	
Halaman Pendaftaran	Button Tambah Kode	Berjalan dengan Baik	✓	
	Button Simpan	Berjalan dengan Baik	✓	
	Button Ubah	Berjalan dengan Baik	✓	
	Button Bersih	Berjalan dengan Baik	✓	
	Button Hapus	Berjalan dengan Baik	✓	
	Button Keluar	Berjalan dengan Baik	✓	

Halaman	Button Simpan	Berjalan dengan Baik	√	
Data Siswa	Button Ubah	Berjalan dengan Baik	√	
	Button Bersih	Berjalan dengan Baik	√	
	Button Hapus	Berjalan dengan Baik	√	
	Button Keluar	Berjalan dengan Baik	√	
Halaman	Button Simpan	Berjalan dengan Baik	√	
Data Kelas	Button Ubah	Berjalan dengan Baik	√	
	Button Bersih	Berjalan dengan Baik	√	
	Button Hapus	Berjalan dengan Baik	√	
	Button Keluar	Berjalan dengan Baik	√	
Halaman	Button Tambah	Berjalan dengan Baik	√	
Pembayaran	Kode Button			
	Sisa Button	Berjalan dengan Baik	√	
	Print Button	Berjalan dengan Baik	√	
	Simpan Button	Berjalan dengan Baik	√	
	Ubah Button	Berjalan dengan Baik	√	
	Bersih Button	Berjalan dengan Baik	√	
	Hapus Button	Berjalan dengan Baik	√	
	Keluar	Berjalan dengan Baik	√	
Halaman	Report Pendaftaran	Berjalan dengan Baik	√	
Laporan	Report Data Siswa			
	Report Data Kelas	Berjalan dengan Baik	√	
	Report	Berjalan dengan Baik	√	
	Pembayaran	Berjalan dengan Baik	√	
Halaman	Tentang informasi	Berjalan dengan Baik	√	
Tentang	pembuat aplikasi			
Halaman	Button Keluar	Berjalan dengan Baik	√	
Keluar				

## V. PENUTUP

### Kesimpulan

Berdasarkan dari hasil penelitian, perancangan aplikasi sampai tahap penyelesaian aplikasi di Ghibrant English Course Pandeglang, maka penulis dapat mengambil kesimpulan adalah Aplikasi sistem informasi administrasi pendaftaran ghibrant dirancang dengan pemodelan *Unified Modeling Language (UML)* yang terdiri dari *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, *Class Diagram*, dibuat dengan menggunakan Borland Delphi7 dan *database MySQL* yang dapat digunakan di Windows 7, Windows 8, Windows 10. Sehingga membuat proses administrasi pendaftaran di ghibrant Pandeglang menjadi lebih praktis dan optimal. Dengan adanya aplikasi sistem informasi ghibrant English course ini dapat mempermudah pengolahan data administrasi pendaftaran. Sehingga

dapat menghasilkan informasi yang cukup akurat dan tidak banyak melakukan kesalahan dalam pengolahan data administrasi.

### Saran

Saran yang diajukan untuk pengembangan aplikasi ini yaitu:

1. Aplikasi ini masih bersifat *offline* sehingga untuk pengembangan selanjutnya dapat diharapkan dijadikan *online*.
2. Penelitian ini masih ada kekurangannya sehingga perlu ditingkatkan lagi pengembangannya di masa yang akan datang agar sistem tersebut semakin bagus dan lebih baik lagi.

## DAFTAR PUSTAKA

- Anggraeny, Krisanti Vonny dan Wellia Shinta Sari (2015) Melakukan Penelitian Tentang “Sistem Informasi Administrasi Pada Bimbingan Tes Alumni 70 Jakarta”
- Al Fatta, H., (2007), Analisis dan Perancangan Sistem Informasi untuk Keunggulan Bersaing Perusahaan & Organisasi Modern, ANDI : Yogyakarta.
- A.S. Rosa, M. Shalahuddin, (2011), Modul Pembelajaran Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek), MODULA : Bandung.
- Irmayani, Deci (2014) Melakukan Penelitian Tentang “Sistem Informasi Administrasi Kesiswaan Pada Smk Pemda Rantauprapt Berbasis Client Server”. Medan.
- Meilani Elly, Heny Sumarti, Mardiani dan Della Oktaviani (2014) Melakukan Penelitian Tentang “Sistem Informasi Administrasi Dan Pembelajaran Siswa Berbasis Desktop Dan Website Pada Family Course” Bandung.
- Melissa, Ajeng (2013) Melakukan Penelitian Tentang “Sistem Informasi Administrasi Pendidikan Pada Bimbingan Belajar Super Student Club (SSC) Ngaliyan Semarang”.
- Sutabri, T., 2012, Analisis Sistem Informasi, Andi: Yogyakarta.
- Waljiyanto, 2003, Sistem Basis Data, Graha Ilmu: Yogyakarta.
- Yakub, 2012, Pengantar Sistem Informasi, Graha Ilmu: Yogyakarta.