

BAB II

PEMBAHASAN

2.1 Teori pendukung

Landasan teori/kajian pustaka yang penulis pakai dalam pembuatan sistem adalah sebagai berikut :

1) Manajemen Data

Manajemen data adalah bidang studi yang luas, tapi pada dasarnya adalah proses pengelolaan data sebagai sumber daya yang berharga bagi suatu organisasi atau bisnis. Manajemen Data merupakan proses pengembangan arsitektur data, praktek dan prosedur yang berhubungan dengan data dan kemudian mengeksekusi aspek-aspek ini secara teratur. Aspek yang penting dalam manajemen data adalah kelengkapan identitas data.

Manajemen Data yang merupakan bagian dari manajemen sumber daya informasi mencakup semua kegiatan yang memastikan bahwa data:

- Data Akurat
- Up to Date (Mutakhir)
- Aman
- Tersedia bagi pemakai (user)

2.1.2 UML (*Unified Modelling Language*)

Menurut Munawar (2005, hal 17) *Unified Modelling language* (UML) adalah suatu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi objek, UML menyediakan pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi dan mengkomunikasikan rancangan mereka dengan yang lain.

Dengan menggunakan UML, dapat membuat model untuk semua jenis aplikasi peranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, dan dapat digunakan juga untuk mendefinisikan notasi dan *syntax*. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram-diagram piranti lunak, setiap bentuk mempunyai makna tertentu dan UML *syntax* mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan.

UML adalah salah satu bentuk notasi atau bahasa yang sama yang digunakan oleh professional dibidang *software* untuk menggambarkan atau memodelkan sebuah sistem *software*. Sebelumnya ada banyak notasi atau bahasa lain untuk mencapai keperluan yang sama misalnya DFD (*Data Flow Diagram*). Tetapi sejak matang dan populernya teknologi pemrograman, perancangan, dan analisis berorientasi objek, UML telah menjadi *de facto standard language*.

Diagram-diagram yang terdapat dalam UML antara lain: *use case diagram*, *class diagram*, *statechart diagram*, *activity diagram*, *sequence diagram*, *collaboration diagram*, *component diagram*, *deployment diagram*.

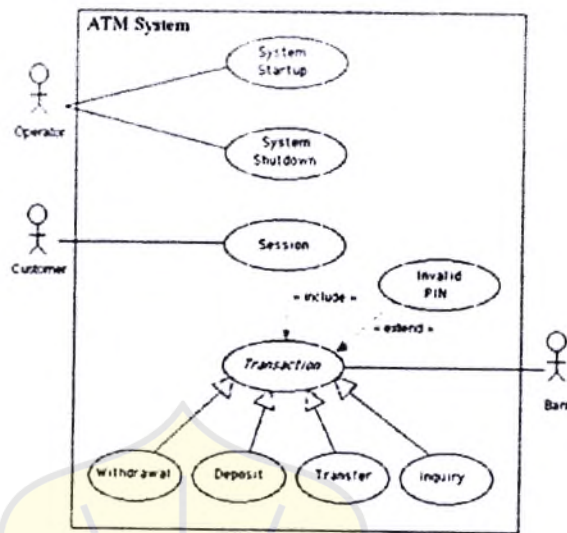
2.1.2.1 Use Case Diagram

Munawar (2005, hal 109-100) menjelaskan *Use case diagram* merupakan deskripsi fungsi dari sebuah sistem perspektif pengguna. *Use case* bekerja dengan cara memdeskripsikan tipikal interaksi antar *user* (pengguna) sebuah sistem dengan sistem-nya sendiri bagaimana sistem tersebut digunakan atau “apa” yang diperbuat sistem dan bukan “bagaimana”. *Use case* dapat dipresentasikan sebuah interaksi antara *actor* dengan sistem .

Use case Diagram dapat sangat membantu bila sedang menyusun *requirement* (Jacobsoonet all, 1992). Diagram ini menunjukkan tiga aspek dari sistem yaitu *actor*, *use case*, dan sistem atau subsistem.

Dilihat dari segi pandang Romi Satria Wahono dan Sri Dharyanti sebuah *use case* dapat meng-include fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di-include akan dipanggil setiap kali *use case* yang meng-include dieksekusi secara normal. Sebuah *use case* dapat di-include oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang *common*. Sebuah *use case* juga dapat meng-extend *use case* lain dengan *behaviour*-nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain.

Contoh Use Case Diagram



Gambar 2.1 Contoh Use Case Diagram

2.1.2.2 Activity Diagram

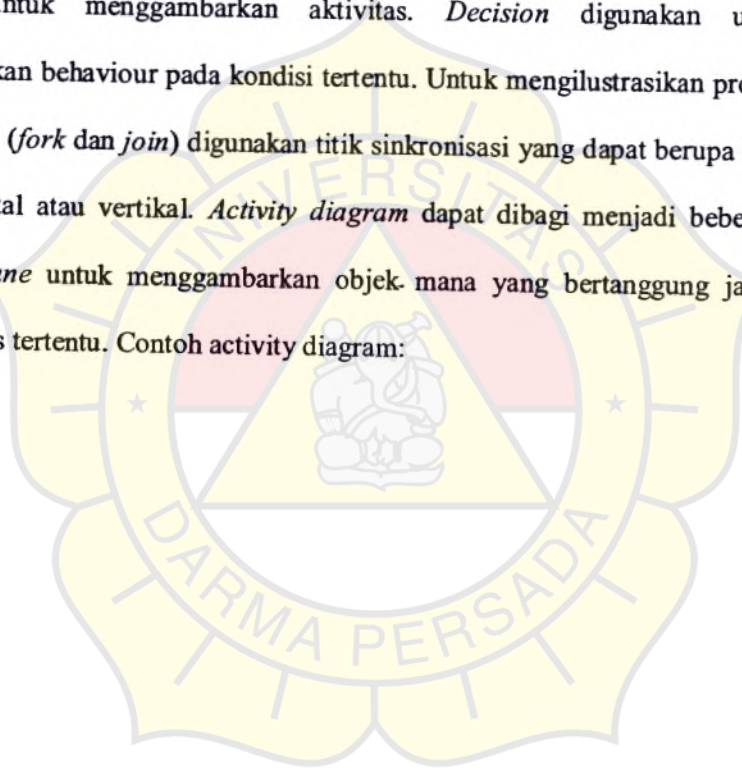
Romi Satria Wahono mengatakan bahwa *Activity diagrams* menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

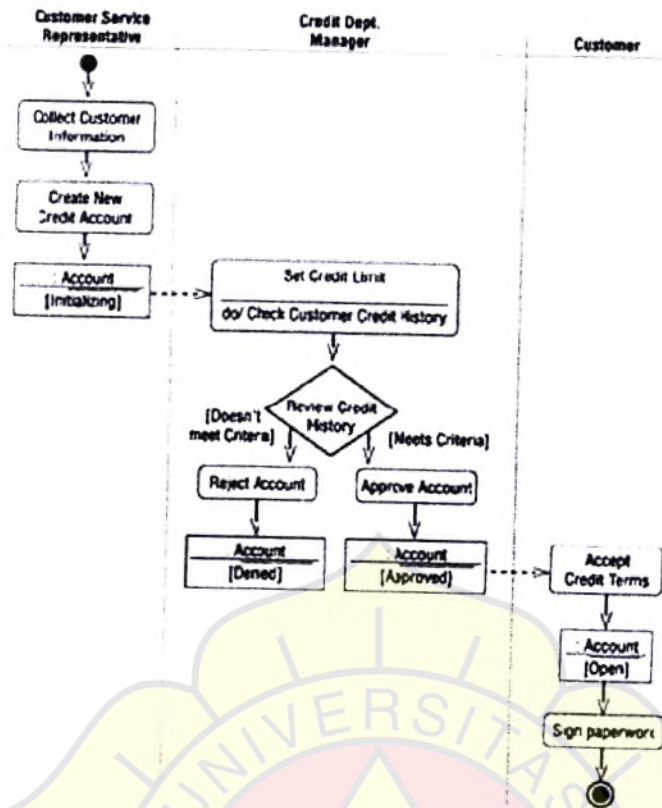
Activity diagram merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem)

secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas.

Sama seperti *state*, standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. *Decision* digunakan untuk menggambarkan behaviour pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (*fork* dan *join*) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal. *Activity diagram* dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu. Contoh activity diagram:





Gambar 2.2 Contoh Activity Diagram

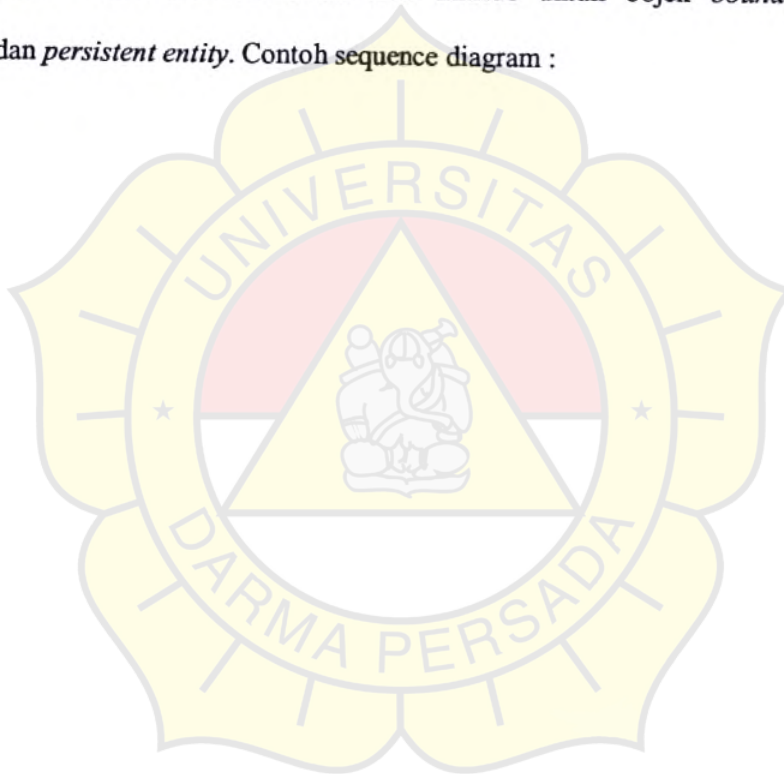
2.1.2.3 Sequence Diagram

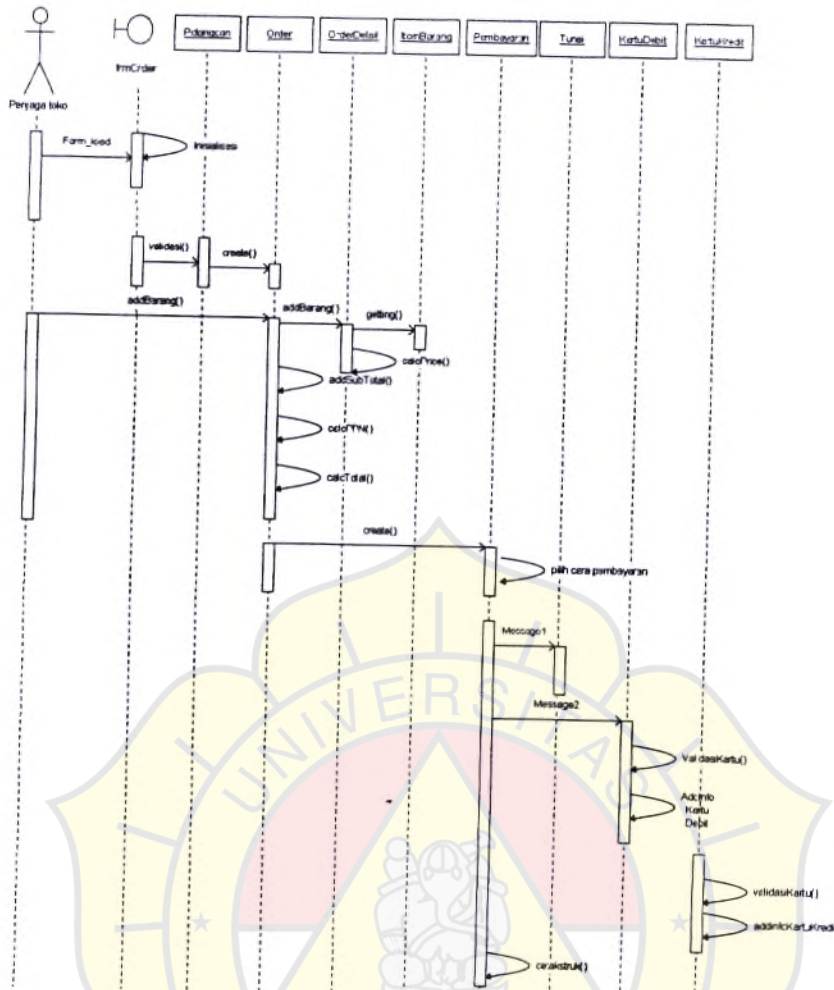
Sequence diagram dilihat dari kacamata Romi Satria Wahono dan Sri Dharyanti menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

Munawar mengatakan *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa

yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan.

Masing-masing objek, termasuk aktor, memiliki *lifeline* vertikal. *Message* digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, *message* akan dipetakan menjadi operasi/metoda dari *class*. *Activation bar* menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah *message*. Untuk objek-objek yang memiliki sifat khusus, standar UML mendefinisikan *icon* khusus untuk objek *boundary*, *controller* dan *persistent entity*. Contoh *sequence diagram* :



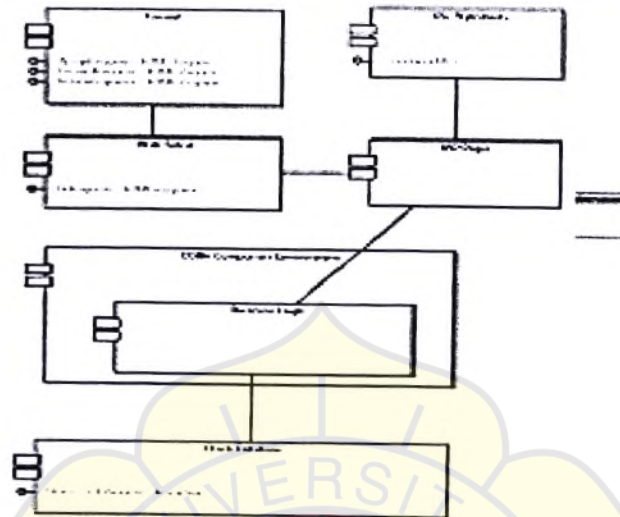


Gambar 2.3 Contoh Sequence Diagram

2.1.2.4 Component Diagram

Component diagram menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency*) di antaranya. Komponen piranti lunak adalah modul berisi *code*, baik berisi *source code* maupun *binary code*, baik *library* maupun *executable*, baik yang muncul pada *compile time*, *link time*, maupun *run time*. Umumnya komponen terbentuk dari

beberapa *class* dan/atau *package*, tapi dapat juga dari komponen-komponen yang lebih kecil. Komponen dapat juga berupa *interface*, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain. Contoh component diagram:



Gambar 2.4 Contoh Component Diagram

2.1.3 Tools Pemrograman

2.1.3.1 HTML

HTML (*Hyper Text Markup Language*) adalah sebuah *bahasa markup* yang digunakan untuk membuat sebuah halaman web dan menampilkan berbagai informasi di dalam sebuah browser Internet.

HTML berupa kode-kode tag yang menginstruksikan browser untuk menghasilkan tampilan sesuai dengan yang diinginkan. Sebuah file yang merupakan file HTML dapat dibuka dengan menggunakan browser web seperti Mozilla Firefox atau Microsoft Internet Explorer. HTML juga dapat dikenali oleh

aplikasi pembuka email ataupun dari PDA dan program lain yang memiliki kemampuan browser.

2.1.3.2 CSS

CSS = *Cascading Style Sheets* (Bahasa lembar Gaya). CSS merupakan bahasa yang digunakan untuk mengatur tampilan suatu dokumen yang ditulis dalam bahasa markup / markup language. Jika kita berbicara dalam konteks web, bisa di artikan secara bebas sebagai : CSS merupakan bahasa yang digunakan untuk mengatur tampilan / desain suatu halaman HTML.

Dulu, sebelum CSS menjadi standar untuk mendesain halaman web seperti sekarang, halaman web di desain menggunakan `<table>`. jadi dibuat dulu desainnya, dalam format .psd atau jpeg, lalu di slice atau di potong potong menjadi bagian – bagian terpisah. setelah itu dibuat table dengan ukuran yang sesuai, lalu desain tadi di “tempel” pada table sebagai image yang melekat pada tabel, atau sebagai background. Kelemahan dari cara seperti ini adalah, halaman web menjadi berat karena kolom – kolom penyusun table `<tr>` dan `<td>` diberi tambahan atribut `image source`.

2.1.3.3 Java Script

JavaScript adalah bahasa pemrograman berbasis prototipe yang berjalan disisi klien. Jika kita berbicara dalam konteks web, sederhananya, kita dapat memahami JavaScript sebagai bahasa pemrograman yang berjalan khusus untuk di browser atau halaman web agar halaman web menjadi lebih hidup. Kalau dilihat dari suku katanya terdiri dari dua suku kata, yaitu Java dan Script. Java

adalah Bahasa pemrograman berorientasi objek, sedangkan Script adalah serangkaian instruksi program.

JavaScript bekerja pada sisi browser. maksudnya begini : untuk menampilkan halaman web, user menuliskan alamat web di address bar url. setelah itu, browser “mengambil” file html (dengan file jJavaScript yang melekat padanya jika memang ada) ke server yang beralamat di URL yang diketikan oleh user. Selesai file diambil, file ditampilkan pada browser. Nah, setelah file JavaScript berada pada browser, barulah script JavaScript tersebut bekerja.

Efek dari Javascript yang bekerja pada sisi browser ini, Javascript dapat merespon perintah user dengan cepat, dan membuat halaman web menjadi lebih responsif. JavaScript melakukan apa yang tidak bisa dilakukan oleh HTML, PHP, dan CSS : menangani hal – hal yang membutuhkan respons cepat terhadap aksi dari user.

Implementasi terpopuler saat ini dari pemrograman JavaScript adalah teknik AJAX. (Asynchronous JavaScript and XMLHTTP). teknik ini sering digunakan oleh aplikasi berbasis web seperti Gmail, Google Reader, dan lain lain. Teknik yang membuat pertukaran data antara server dan browser terjadi di belakang layar sehingga interaksi antara user dan aplikasi web semakin responsif. Post tersendiri untuk membahas hal ini akan kita siapkan.

2.1.3.4 Bahasa Pemrograman PHP (*Personal Home Page*)

PHP singkatan dari PHP *Hypertext Preprocessor* yaitu bahasa pemrograman *web server-side* yang bersifat *open source*. PHP merupakan *script* yang menyatu dengan HTML dan berada pada *server* (*server side HTML embedded scripting*). PHP adalah *script* yang digunakan untuk membuat halaman *web* yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh *client*. Mekanisme ini menyebabkan informasi yang diterima *client* selalu yang terbaru/*up to date*. Semua *script* PHP dieksekusi pada *server* dimana *script* tersebut dijalankan.

2.1.3.5 Database 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.

Sebagai server database dengan konsep database modern, MySQL memiliki keistimewaan. Beberapa keistimewaan dimiliki MySQL sebagai berikut :

- 1). *Portability* : Database MySQL berfungsi dengan stabil tanpa kendala, berarti berlaku pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac OS X Server, Solaris, Amiga, HP-Uinx, dan lain-lain.
- 2). *Open Source* : MySQL merupakan *database open source* (gratis), di bawah lisensi GPL sehingga dapat memperoleh dan menggunakannya secara cuma-cuma tanpa membayar sepersen pun.
- 3). *Multiuser* : MySQL merupakan *database* yang dapat digunakan untuk

menangani beberapa *user* dalam waktu bersamaan tanpa mengalami masalah. Dan memungkinkan sebuah *database server* MySQL dapat diakses *client* secara bersamaan pula.

- 4). *Performace Tuning* : MySQL mempunyai kecepatan yang cukup baik dalam menangani query-query sederhana, serta mampu memproses lebih banyak SQL per satuan waktu.
- 5). *Column Type* : Database MySQL didukung dengan tipe data yang sangat kompleks, seperti *signed/unsigned integer, float, double, char, varchar, text, blob, data, time, datetime, timestamp, year, set* serta *enum*.
- 6). *Command dan Functions* : MySQL server memiliki operator dan fungsi secara penuh yang mendukung perintah SELECT dan WHERE dalam query.
- 7). *Security* : Sistem *Security* pada MySQL mempunyai beberapa lapisan sekuritas seperti tingkatan *subnetmask, hostname*, dan izin akses *user* dengan sistem perizinan yang mendetil serta password terenkripsi.
- 8) *Scalability dan Limits* : MySQL mempunyai kemampuan menangani database dalam skala cukup besar, dengan jumlah *record* lebih dari 50 juta dan 60 ribu tabel serta 5 miliar baris. Selain itu dapat menampung indeks sampai 32 indeks pada tiap tabelnya.
- 9). *Connectivity* : Adanya kemampuan MySQL melakukan koneksi dengan *client* menggunakan protokol TCP/IP, Unix socket (Unix), atau Named Pipes (NT).
- 10). *Localization* : Adanya kemampuan dalam mendeteksi kesalahan (*error code*) pada *client* menggunakan lebih dari dua puluh bahasa.
- 11). *Interface* : MySQL memiliki *interface* terhadap berbagai aplikasi dan bahasa pemrograman menggunakan fungsi API (*Application Programming Interface*).

12). *Clients dan Tools* : Database MySQL dilengkapi berbagai tools yang dapat digunakan untuk administrasi database.

13). *Struktur Tabel* : MySQL memiliki struktur tabel cukup baik serta cukup fleksibel, misalnya ketika menangani alter table.

2.1.3.6 XAMPP

XAMPP adalah perangkat lunak gratis, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai server yang berdiri sendiri (*localhost*), yang terdiri atas program Apache HTTP Server, MySQL database, dan penterjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP dan Perl. Program ini tersedia dalam GNU General Public Lisensi dan bebas, merupakan *web server* yang mudah digunakan yang dapat melayani tampilan halaman *web* yang dinamis. Untuk mendapatkannya dapat mendownload langsung dari *web* resminya.

Xampp merupakan sebuah *tool* yang menyediakan paket perangkat lunak ke dalam satu buah paket. Dengan menginstall XAMPP maka tidak perlu lagi melakukan instalasi dan konfigurasi web server Apache, PHP dan MySQL secara manual. XAMPP akan menginstalasi dan mengkonfigurasikannya secara otomatis untuk anda atau auto konfigurasi.

2.2 Tinjauan Kasus

Saat ini bagian Tata Usaha yang ada pada Sekolah dasar negeri masih menggunakan sistem yang manual dalam menangani proses administrasi,

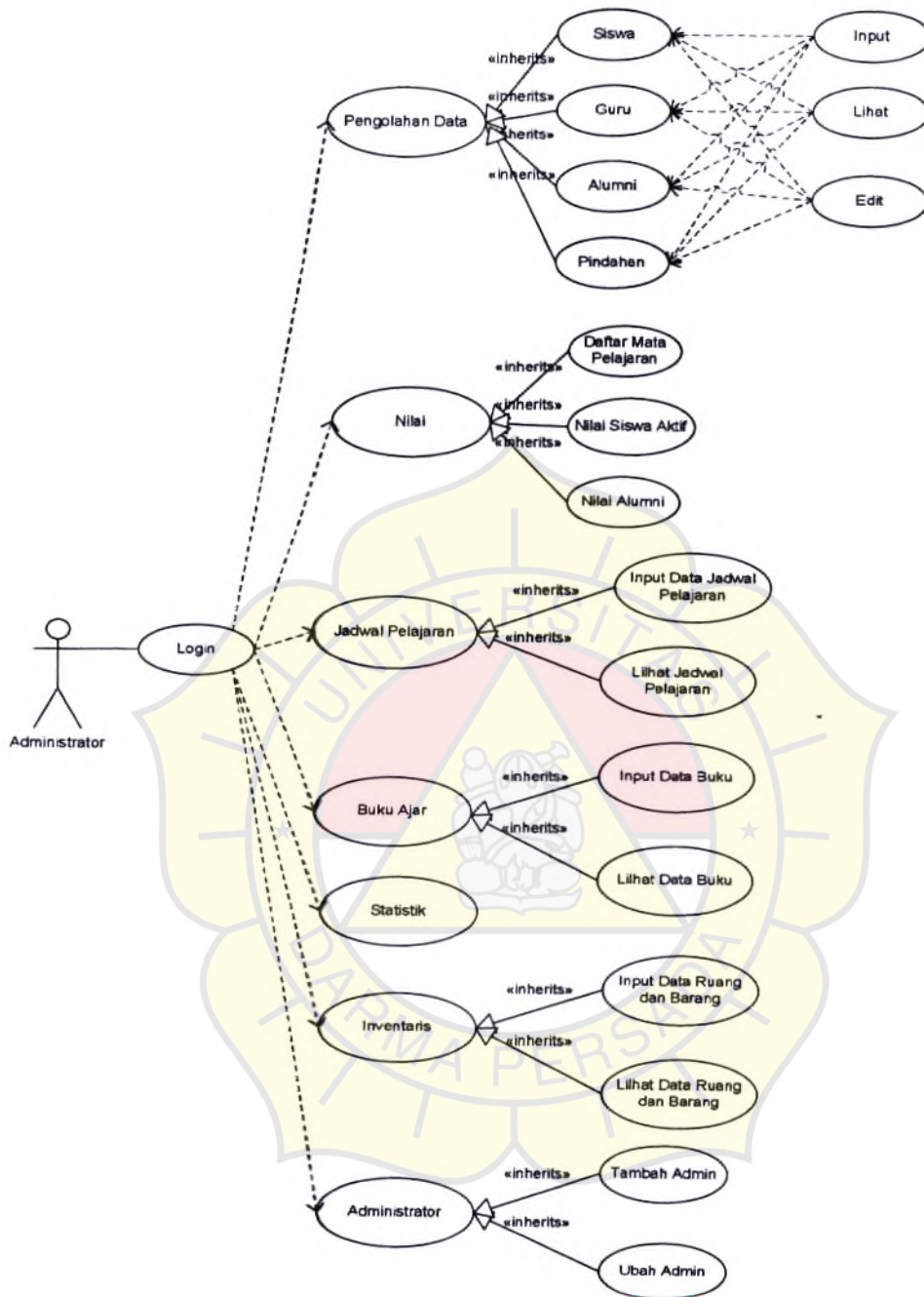
sehingga pada prakteknya ketika dibutuhkan informasi maka bagian tata usaha mengalami kendala seperti harus mengumpulkan dokumen-dokumen yang dibutuhkan sehingga dalam menyajikan informasi masih memerlukan waktu yang lama. Maka dari itu penulis berusaha membuat Aplikasi Pengolahan Data Pada Sekolah Dasar Negeri sehingga dalam proses administrasi dan pengolahan data dapat menjadi mudah.

2.3 Spesifikasi Rancangan Program

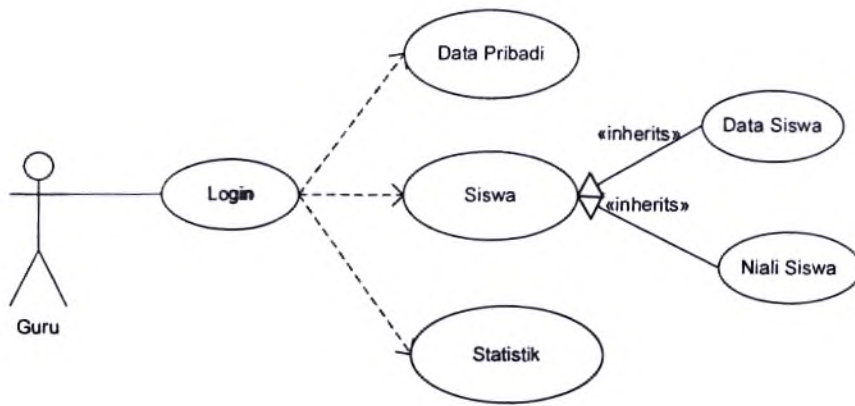
Spesifikasi rancangan program yang digunakan dalam penyusunan Tugas Akhir ini adalah sebagai berikut :



2.3.1 Use Case Diagram:

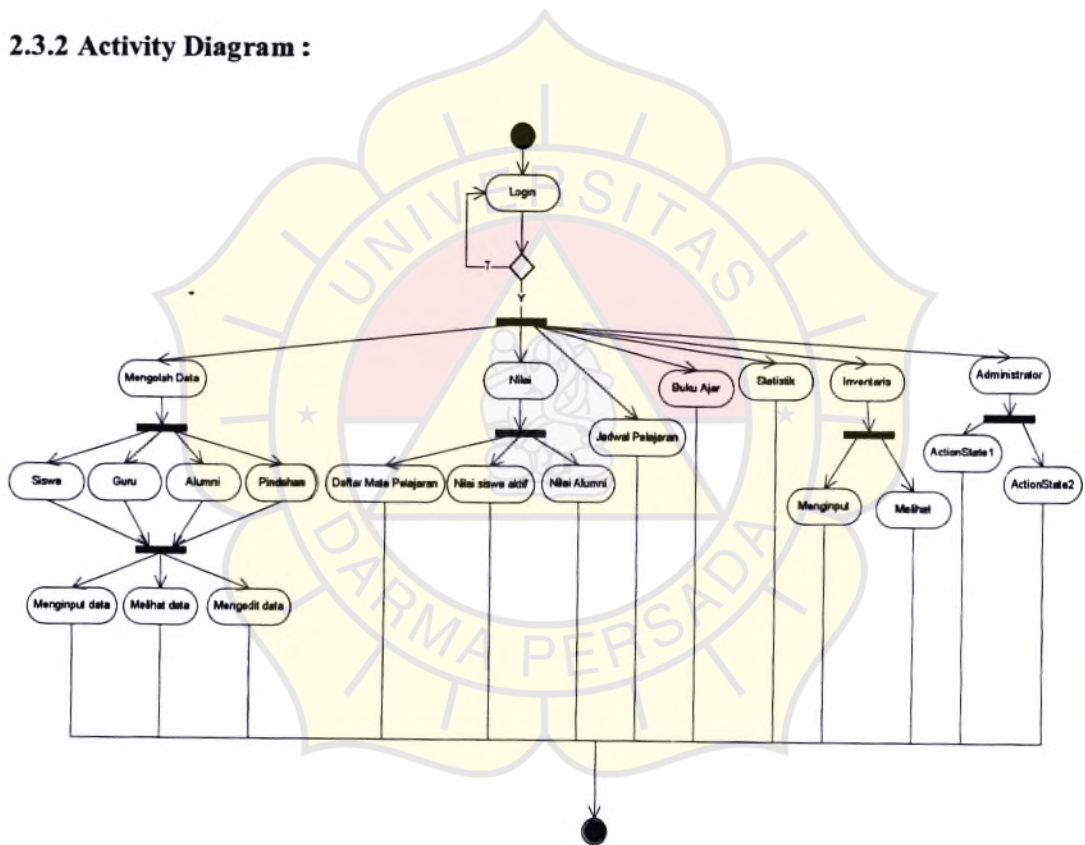


Gambar 2.5 Use Case Diagram Administrator

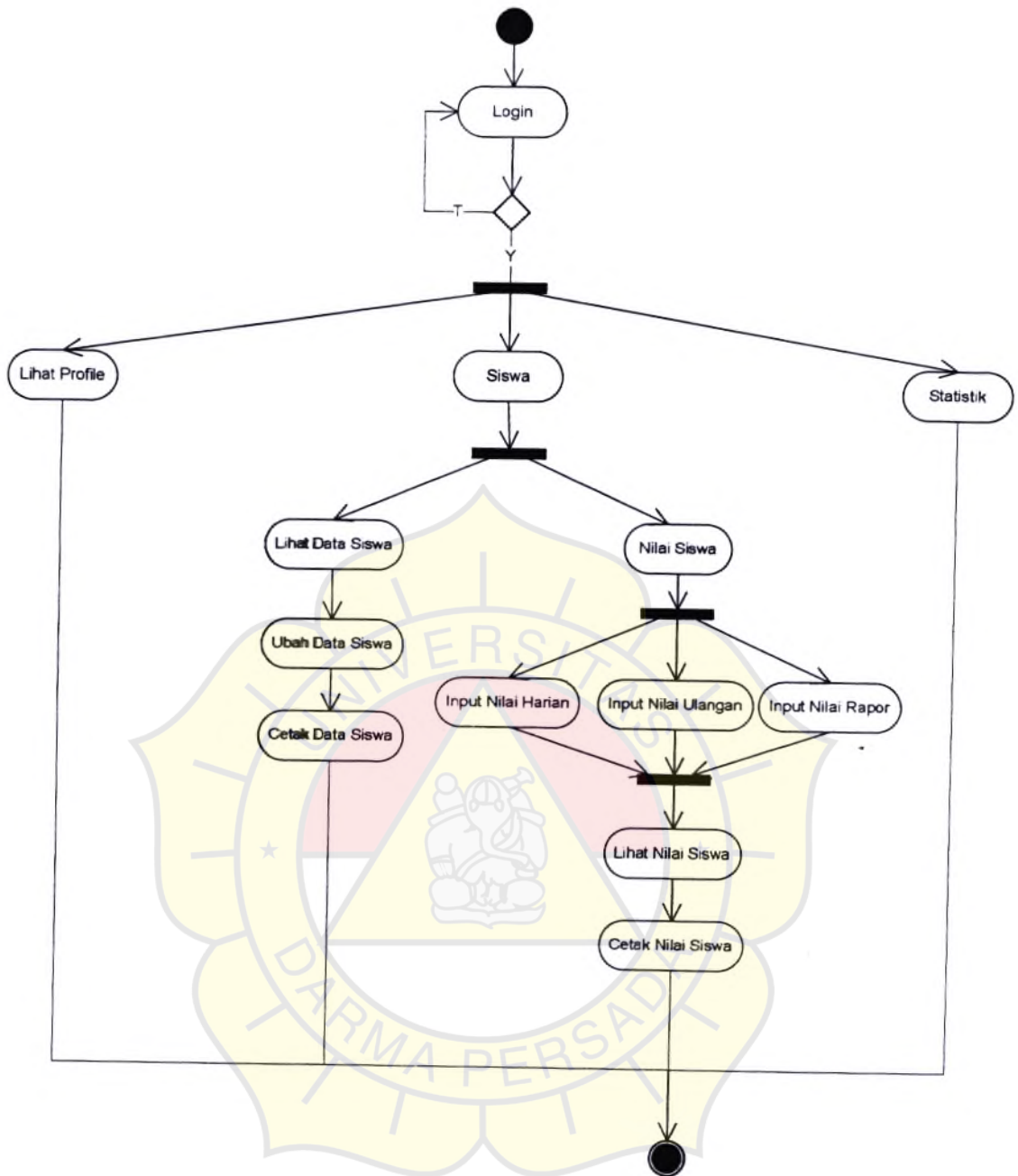


Gambar 2.6 Use Case Diagram Walikelas

2.3.2 Activity Diagram :

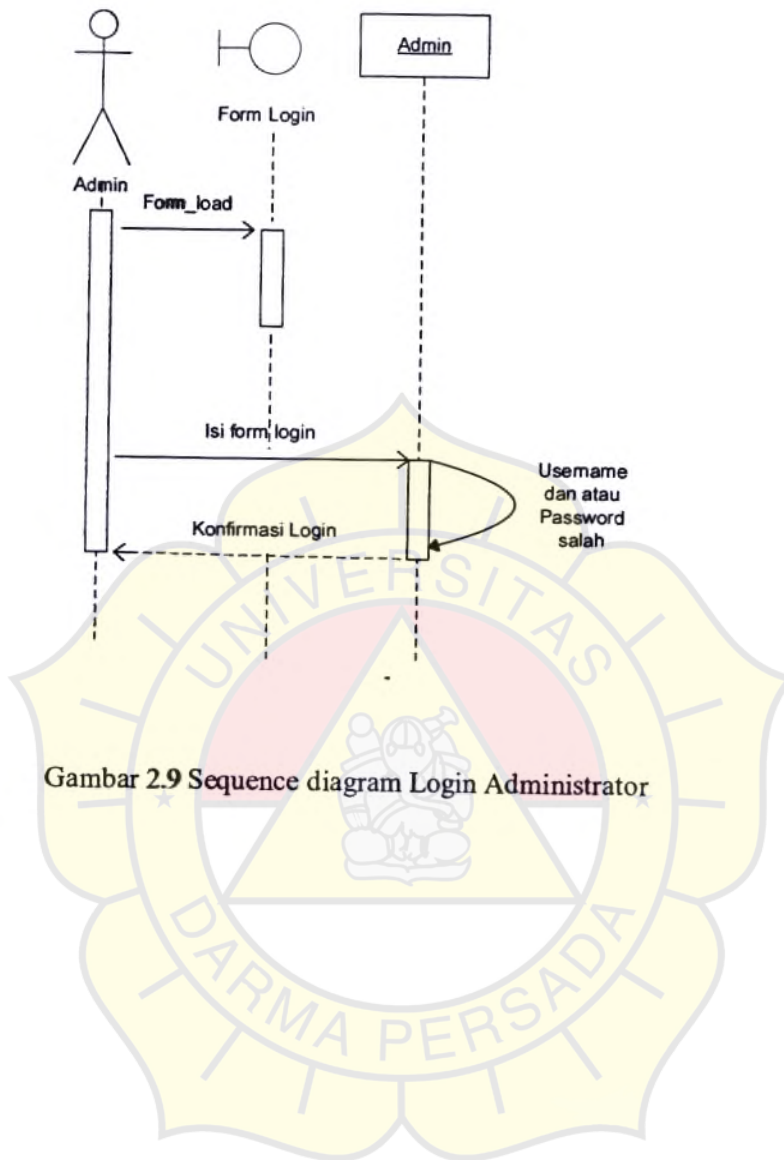


Gambar 2.7 Activity Diagram Administrator

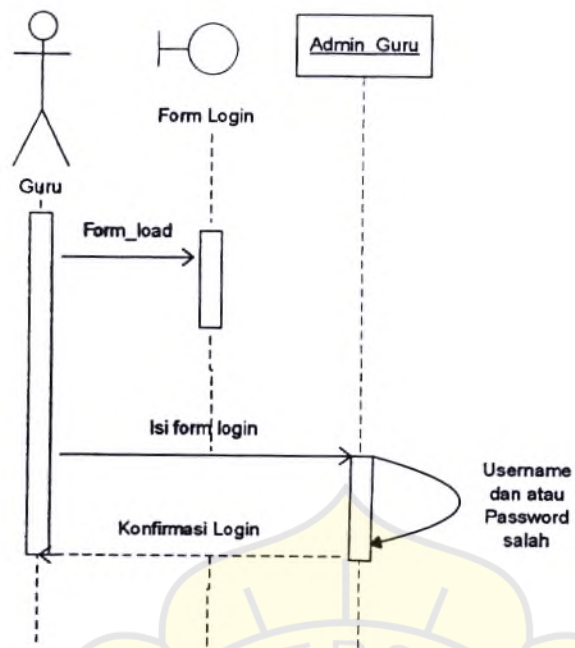


Gambar 2.8 Activity Diagram Walikelas

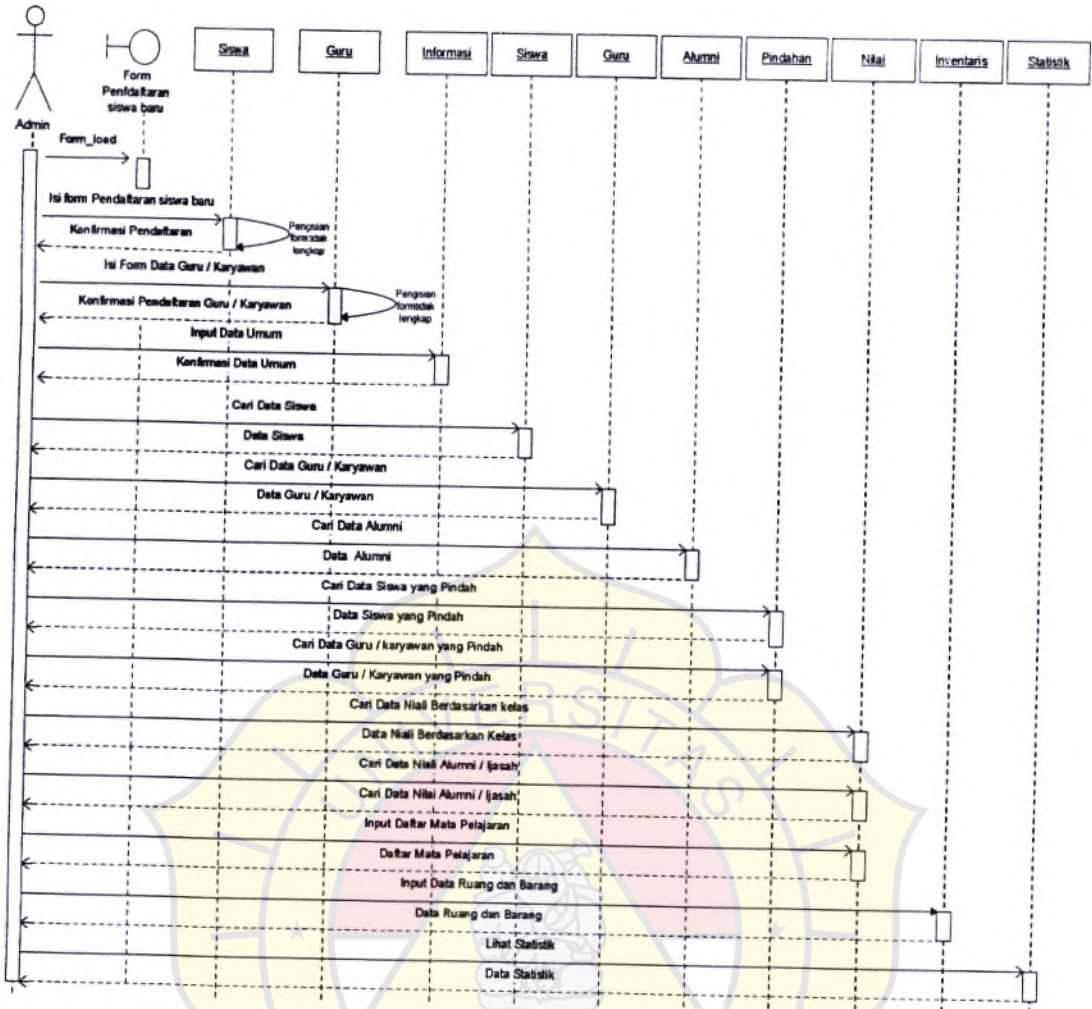
2.3.3 Sequence Diagram :



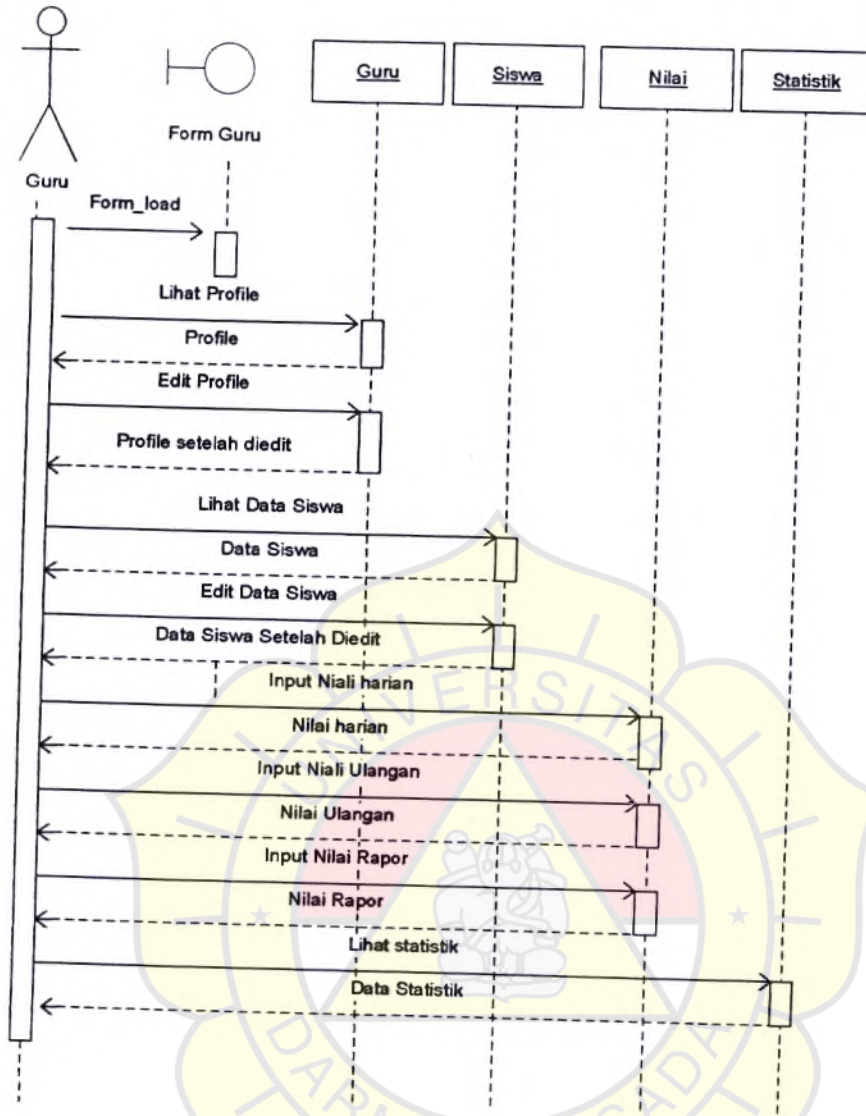
Gambar 2.9 Sequence diagram Login Administrator



Gambar 2.10 Sequence Diagram Login Walikelas

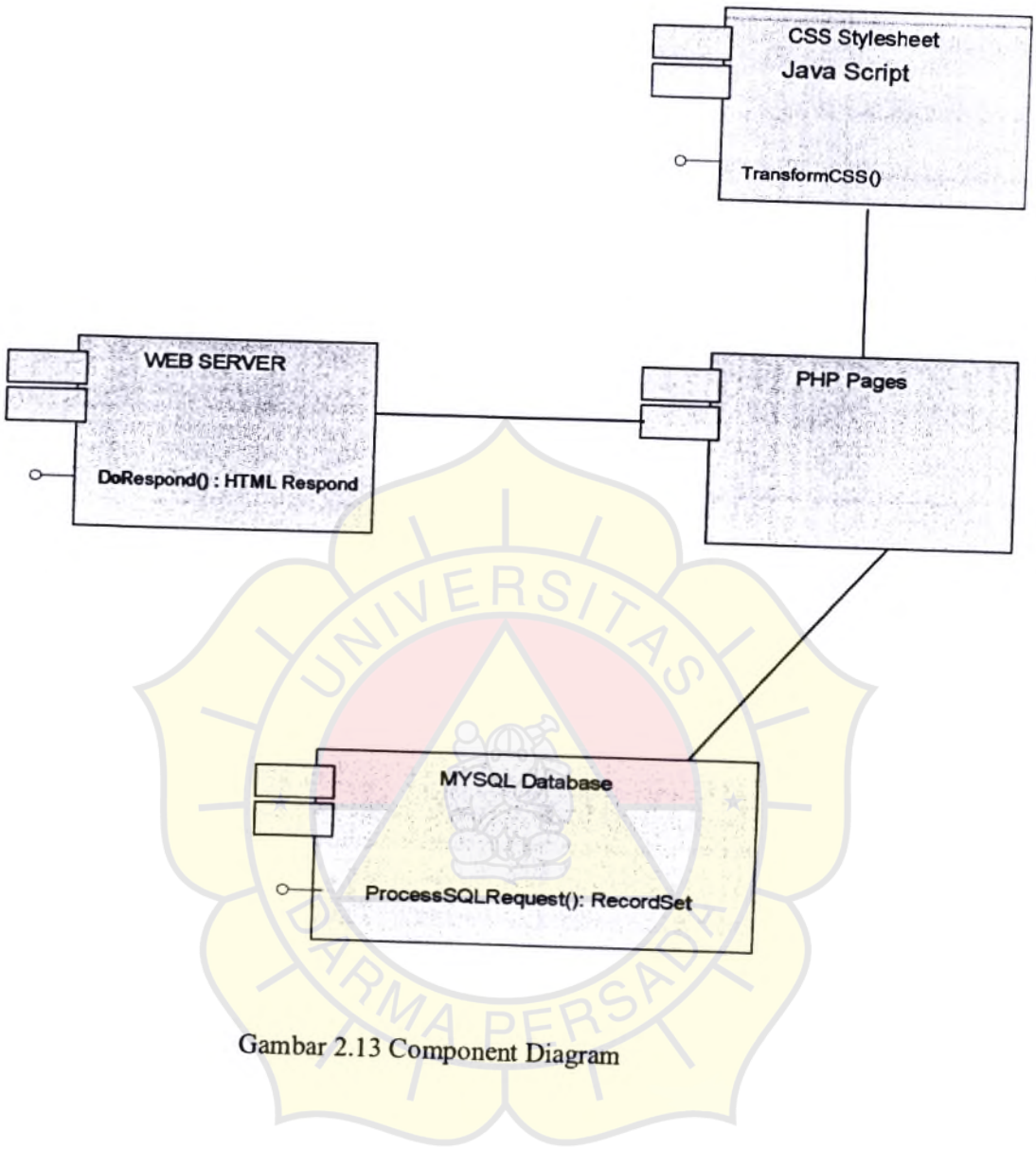


Gambar 2.11 Sequence Diagram Administrator



Gambar 2.12 Sequence Diagram Walikelas

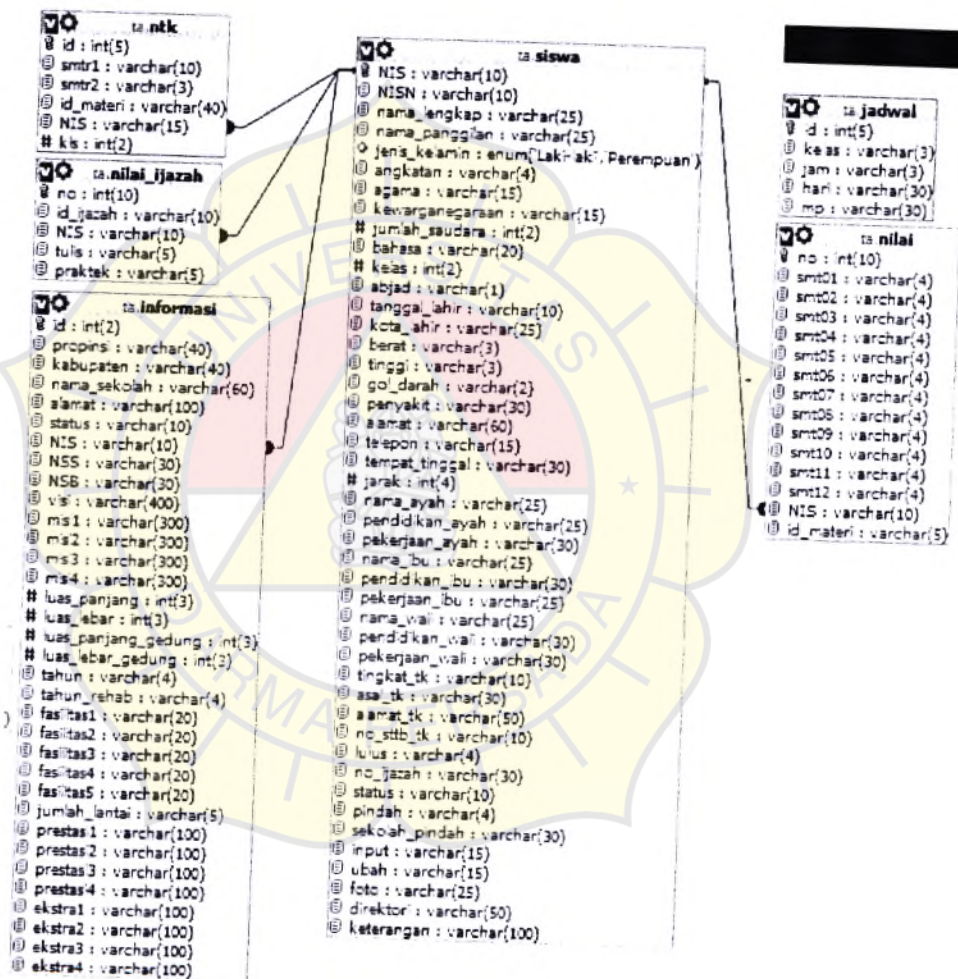
2.3.4 Component Diagram



Gambar 2.13 Component Diagram

2.3.5 Tabel Relasi

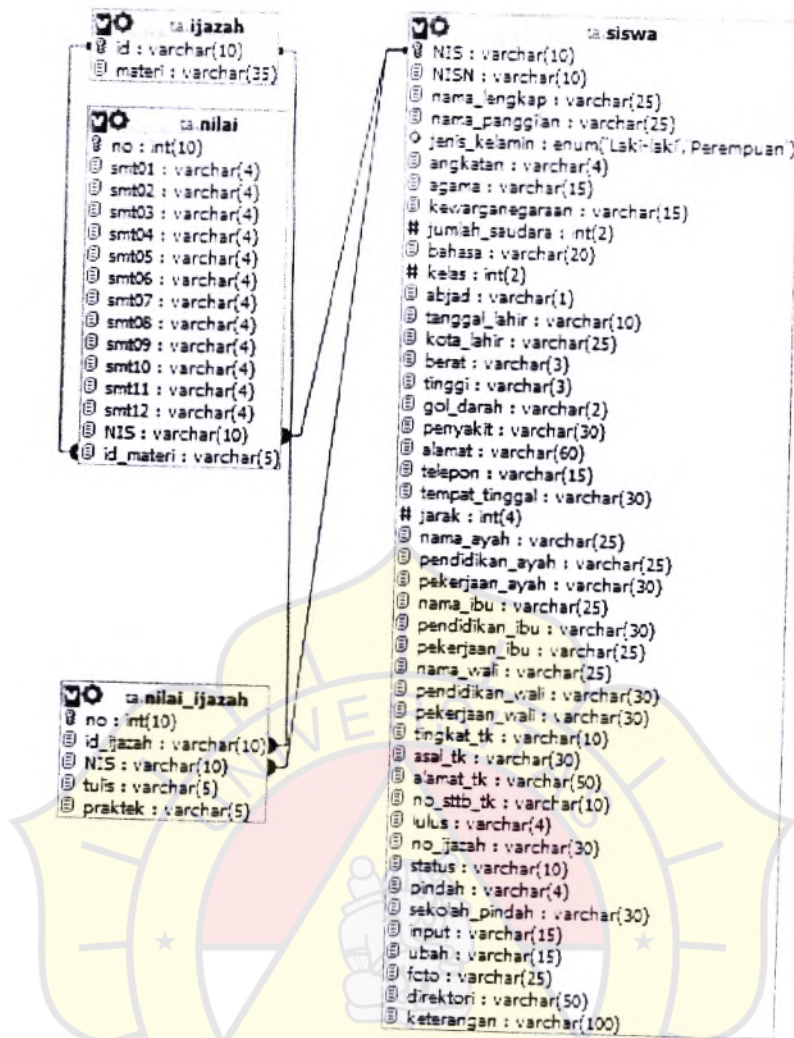
Relasi adalah hubungan yang terjadi antara satu entitas dengan entitas lain atau hubungan yang terjadi antara 2 tabel atau lebih. Dalam membangun sebuah aplikasi database, relasi adalah suatu yang pasti akan dilakukan guna mendapatkan informasi yang mungkin terdapat pada table lain. Berikut adalah Relasi yang terdapat dalam database :



Gambar 2.13 Relasi Tabel Siswa dengan Tabel lain



Gambar 2.13 Gambar Relasi Tabel Guru



Gambar 2.13 Relasi Tabel Nilai dan Ijazah

2.3.6 Rancangan Bentuk Masukan

Rancangan bentuk masukan merupakan desain *interface form-form* yang digunakan untuk pemasukan data. Berikut adalah rancangan layout aplikasi :

HEADER	
INPUT DATA	
DATA SISWA	Pendaftaran Siswa Baru
DATA GURU	Input Data Guru / Karyawan
DATA ALUMNI	Input Data Umum
DATA PINDAHAN	
DATA NILAI	
INVENTARIS	
PROFIL	
STATISTIK	
ADMINISTRATOR	
LOGOUT	
FOOTER	

Gambar 2.14 Tampilan pada Halaman Input Data

HEADER	
INPUT DATA	
DATA SISWA	Data Berdasarkan Kelas
DATA GURU	Data Berdasarkan Angkatan
DATA ALUMNI	
DATA PINDAHAN	Cari Berdasarkan Nama : <input type="text"/> <input type="text"/> OK
DATA NILAI	
INVENTARIS	Cari Berdasarkan NIS : <input type="text"/> <input type="text"/> OK
PROFIL	
STATISTIK	
ADMINISTRATOR	
LOGOUT	
FOOTER	

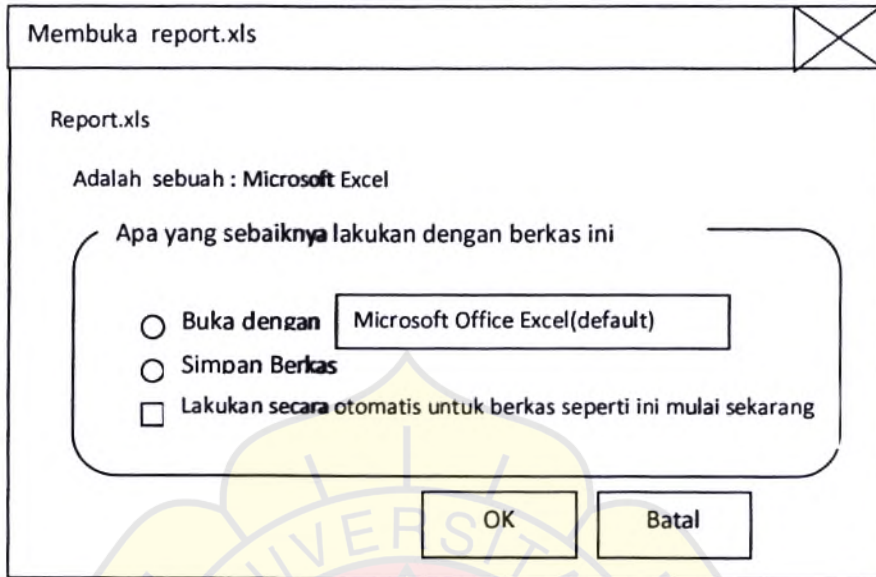
Gambar 2.14 Tampilan Pada Halaman Data Siswa

HEADER	
INPUT DATA	
DATA SISWA	Daftar Keseluruhan
DATA GURU	Cari Berdasarkan Nama :
DATA ALUMNI	<input type="text"/> <input type="button" value="OK"/>
DATA PINDAHAN	Cari Berdasarkan NIP :
DATA NILAI	<input type="text"/> <input type="button" value="OK"/>
INVENTARIS	Cari Berdasarkan Wali Kelas :
PROFIL	<input type="text"/> <input type="text"/> <input type="button" value="OK"/>
STATISTIK	
ADMINISTRATOR	Pendaftaran Wali Kelas
LOGOUT	
FOOTER	

Gambar 2.14 Tampilan Pada Halaman Data Guru

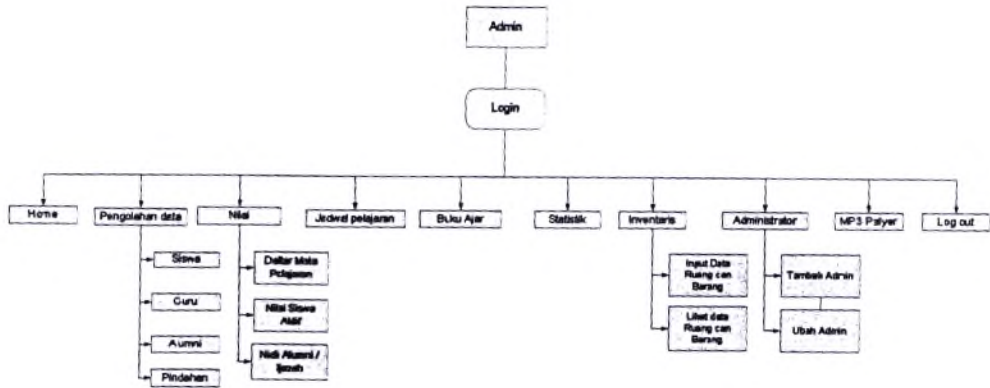
2.3.7 Rancangan Bentuk Keluaran

Download Laporan

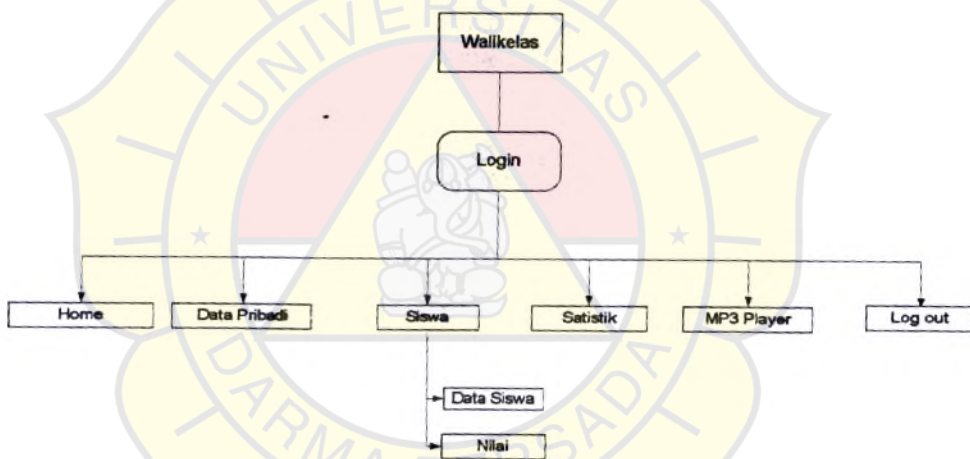


Gambar 2.15 Tampilan Bantuk Keluaran

2.3.8 Spesifikasi Program (HIPO)



Gambar 2.16 HIPO Daigram Administrator



Gambar 2.16 HIPO Daigram Walikelas