

## RANCANG BANGUN APLIKASI PEMBAYARAN SEKOLAH MENGUNAKAN JAVA DAN MySQL BERBASIS *CLIENT SERVER* DI SMA YOS SUDARSO CILACAP

---

Elisa Usada<sup>1</sup>, Anggun Fitriani Isnawati<sup>2</sup>, Dewi Fatimah<sup>3</sup>

<sup>1,2,3</sup>Program Studi Diploma III Teknik Telekomunikasi, Purwokerto

<sup>1</sup>[elisa@akatelsp.ac.id](mailto:elisa@akatelsp.ac.id), <sup>2</sup>[anggun\\_fitriani@yahoo.com](mailto:anggun_fitriani@yahoo.com), <sup>3</sup>[dewiclp\\_d307168@yahoo.co.id](mailto:dewiclp_d307168@yahoo.co.id)

### *Abstrak*

SMA Yos Sudarso merupakan salah satu sekolah yang sudah memanfaatkan komputer untuk menyelesaikan berbagai macam tugas, tetapi pemanfaatannya masih belum maksimal karena ada bagian administrasi yang masih menggunakan metode manual, yaitu pada administrasi pembayaran. Penelitian ini berusaha membuat sebuah aplikasi *client server* berteknologi JAVA dan MySQL untuk mengelola data pembayaran sekolah meliputi pembayaran SPP, pembayaran uang gedung serta pembayaran ujian. Metode *waterfall* digunakan sebagai acuan perancangan dan pengembangan aplikasi. Alat abstraksi sistem yang digunakan adalah *use case diagram* dan *class diagram*. Perancangan basis data dengan menggunakan ERD. Pengujian dilakukan dengan metode *black box* yaitu hanya menguji jalannya fungsi-fungsi yang telah direncanakan tanpa mempedulikan aspek proses internal dalam kode dan algoritma. Pengujian memberikan hasil bahwa aplikasi dapat dijalankan secara *client server* dan fungsi dapat berjalan semestinya. Proses *back up* data otomatis menjadi kekurangan dari aplikasi dalam penelitian ini.

**Kata kunci :** *aplikasi pembayaran, JAVA dan MySQL*

---

### **1. Pendahuluan**

SMA Yos Sudarso merupakan salah satu sekolah yang sudah memanfaatkan komputer untuk menyelesaikan berbagai macam tugas, tetapi pemanfaatannya masih belum maksimal karena ada bagian administrasi yang masih menggunakan metode manual, yaitu pada administrasi pembayaran. Hasil pembayaran dari siswa masih dituliskan secara manual ke dalam buku besar.

Penelitian ini berusaha membuat sebuah aplikasi berteknologi JAVA dan MySQL untuk mengelola data pembayaran sekolah seperti, *input* data siswa, pencarian data siswa, *report* dari pembayaran siswa, serta pencetakan bukti pembayaran. Proses pembayaran yang akan dilibatkan dalam aplikasi adalah pembayaran SPP,

pembayaran uang gedung serta pembayaran ujian. Aplikasi ini akan diterapkan secara *client server* pada komputer yang berada di ruang kepala sekolah, kepala tata usaha dan ruang bendahara. Kepala sekolah dan kepala tata usaha menggunakan aplikasi ini untuk melihat informasi laporan pembayaran jika sewaktu-waktu diperlukan. Bendahara menggunakan aplikasi ini untuk mengelola data pembayaran sekolah seperti, *input* data siswa, pencarian data siswa, *report* dari pembayaran siswa, serta pencetakan bukti pembayaran.

Perancangan dan pembangunan aplikasi dalam penelitian ini menggunakan metodologi *waterfall* dengan tahapan sebagai berikut (Sommerville, 2009):

- a. *Requirement analysis and definition*, mengumpulkan kebutuhan secara lengkap kemudian dianalisis dan didefinisikan kebutuhan yang harus dipenuhi oleh program yang akan dibangun.
- b. *System and software design*, desain dikerjakan setelah kebutuhan selesai dikumpulkan secara lengkap. Alat perancangan yang digunakan adalah UML (*Unified Modelling Language*) untuk merancang objek dalam program, beserta ERD (*Entity Relationship Diagram*) untuk merancang kebutuhan basis data.
- c. *Implementation and unit testing*, desain program diterjemahkan ke dalam bahasa pemrograman yang sudah ditentukan, yaitu JAVA.
- d. *Integration and system testing*, penyatuan unit-unit kemudian diuji secara keseluruhan (*system testing*) menggunakan *emulator*.
- e. *Operation and maintenance*, mengoperasikan program di perangkat yang sesungguhnya.

## 2. Bahasa Pemrograman JAVA dan Netbeans IDE

### 2.1 JAVA

JAVA merupakan bahasa pemrograman yang berorientasi objek dan dapat dijalankan pada berbagai platform sistem operasi. JAVA dikembangkan oleh Sun Microsystem Inc. Perkembangan Java tidak hanya terfokus pada satu sistem operasi, tetapi dikembangkan untuk

berbagai sistem operasi dan bersifat open source. Fitur penting bahasa JAVA adalah (Hariyanto, 2007):

- a. Bahasa sederhana, tidak menyediakan fitur-fitur rumit dan secara otomatis dapat melakukan dealokasi memori.
- b. Bahasa pemrograman berorientasi objek, program dipandang sebagai sekelompok objek yang saling berinteraksi.
- c. Bahasa statically typed, seluruh objek dalam JAVA harus dideklarasikan terlebih dahulu.
- d. Bahasa dikompilasi, kompilasi dilakukan dengan JAVA compiler dan menghasilkan bytecode yang memiliki fungsi serupa dengan kode mesin.
- e. Bahasa yang aman, JAVA menyediakan berbagai teknik pengamanan misalnya dengan peniadaan pointer, penggunaan three layer security model, adanya JAVA verifier dan pembatasan kerja applet.
- f. Independen terhadap platform, tipe data dalam JAVA memiliki ukuran yang sama meskipun dijalankan pada platform yang berbeda-beda.
- g. Multithreading, dapat mempunyai lebih dari satu thread eksekusi pada saat yang sama sehingga memungkinkan program menangani tugas secara konkuren.
- h. Mendukung garbage collector, pemrogram tidak perlu menghapus sendiri objek-objek yang tidak digunakan lagi.

- i. Tangguh, JAVA interpreter memeriksa semua akses sistem yang dilakukan sehingga tidak menyebabkan crash pada sistem.
- j. Mampu diperluas, mendukung native method sehingga fungsi dapat ditulis dalam bahasa lain.
- k. 2.2 Netbeans IDE
- l. NetBeans mengacu pada dua hal, yakni platform untuk pengembangan aplikasi desktop Java dan sebuah IDE (Integrated Development Environment) yang dibangun menggunakan platform NetBeans. Platform NetBeans memungkinkan aplikasi dibangun dari sekumpulan komponen perangkat lunak modular yang disebut 'modul'. Sebuah modul adalah suatu arsip Java (Java archive) yang memuat kelas-kelas Java untuk berinteraksi dengan NetBeans Open API dan file manifestasi yang mengidentifikasinya sebagai modul. Aplikasi yang dibangun dengan modul dapat dikembangkan dengan menambahkan modul baru. Netbeans IDE yang digunakan dalam penelitian ini adalah Netbeans 6.5.1.

### 3. MySQL

MySQL merupakan DBMS (Database Management System) yang bersifat open source yaitu kode yang dipakai untuk membuat MySQL dapat dijalankan secara langsung dalam sistem operasi dan dapat diunduh secara gratis di internet. MySQL

awalnya dibuat oleh perusahaan konsultan bernama TcX yang berlokasi di Swedia. Fitur-fitur dalam MySQL adalah sebagai berikut (Kadir, 2008):

- a. Multiplatform  
MySQL tersedia pada beberapa platform seperti Windows, Linux, Unix, dan lain sebagainya.
- b. Andal, cepat, dan mudah digunakan  
MySQL tergolong sebagai server basis data (server yang melayani permintaan terhadap basis data) yang andal, dapat menangani basis data yang besar dengan kecepatan tinggi, mendukung banyak sekali fungsi untuk mengakses basis data, dan mudah digunakan.
- c. Jaminan keamanan akses  
MySQL mendukung pengamanan basis data dengan berbagai kriteria pengaksesan. Sebagai gambaran, dimungkinkan untuk mengatur user tertentu agar bisa mengakses data yang bersifat rahasia. MySQL juga mendukung konektivitas ke berbagai software, misalnya dengan menggunakan ODBC (Open Database Connectivity).
- d. Dukungan SQL  
MySQL mendukung perintah SQL (Structured Query Language). SQL merupakan standar dalam pengaksesan basis data relasional.

### 4. Client Server

LAN (*Local Area Network*) adalah sejumlah komputer yang dihubungkan bersama di dalam suatu areal tertentu yang

tidak begitu luas, seperti di dalam satu kantor atau gedung. LAN tersusun dari beberapa elemen dasar yaitu *hardware* dan *software*. Salah satu tipe LAN adalah *client-server*.

*Client-server* adalah suatu bentuk arsitektur, dimana *client* adalah perangkat penerima yang akan menampilkan dan menjalankan aplikasi (*software* komputer) dan *server* adalah perangkat yang menyediakan dan bertindak sebagai pengelola aplikasi, data, dan keamanannya. *Server* biasanya terhubung dengan *client* melalui kabel UTP dan sebuah kartu jaringan (*network card*). Kartu jaringan ini biasanya berupa kartu PCI atau ISA.

*Client-server* merujuk kepada cara mendistribusikan aplikasi ke pihak *client* dan pihak *server*. Dalam model *client-server*, sebuah aplikasi dibagi menjadi dua bagian yang terpisah (tetapi masih dalam sebuah kesatuan) yakni komponen *client* dan komponen *server*. Komponen *client* dijalankan pada sebuah *workstation*. Pemakai *workstation* memasukkan data dengan menggunakan teknologi pemrosesan tertentu, kemudian mengirimkannya ke komponen *server*, umumnya berupa permintaan layanan tertentu yang dimiliki oleh *server*. Komponen *server* akan menerima permintaan layanan tersebut dan langsung memprosesnya serta mengembalikan hasil pemrosesan kepada *client*. *Client* pun menerima informasi hasil pemrosesan data tadi dan menampilkannya kepada pemakai

dengan menggunakan aplikasi yang digunakan oleh pemakai.

*Client-server* merupakan penyelesaian masalah pada *software* yang menggunakan basis data sehingga setiap komputer tidak perlu diinstal *database*. Dengan metode *client-server*, *database* dapat diinstal pada komputer *server* dan aplikasinya diinstal pada *client*. Komponen *client* juga sering disebut sebagai *front-end*, sedangkan komponen *server* disebut sebagai *back-end*.

## 5. UML

UML (*Unified Modelling Language*) adalah sebuah bahasa yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menggunakan *class* dan *operation* dalam konsep dasarnya, maka UML lebih cocok untuk penulisan piranti lunak dalam bahasa-bahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk memodelkan aplikasi prosedural dalam VB atau C.

UML terdiri dari diagram-diagram yang merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Diagram UML yang digunakan dalam penelitian ini adalah *use case* dan *class diagram*.

### 5.1 Diagram Use Case

*Use case diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah

sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, menambah sebuah daftar belanja, dan sebagainya. Seorang atau sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.

## 5.2 Diagram Class

*Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek serta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. Jenis hubungan antar *Class* dalam UML adalah sebagai berikut:

- a. Asosiasi, yaitu hubungan statis antar *class*. Umumnya menggambarkan *class* yang memiliki atribut berupa *class* lain, atau *class* yang harus mengetahui eksistensi *class* lain. Panah *navigability* menunjukkan arah *query* antar *class*.
- b. Agregasi, yaitu hubungan yang menyatakan bagian (“terdiri atas..”).
- c. Pewarisan, yaitu hubungan hirarkis antar *class*. *Class* dapat diturunkan dari *class* lain dan mewarisi semua atribut dan metode *class* asalnya dan menambahkan fungsionalitas baru, sehingga ia disebut anak dari *class* yang diwarisinya. Kebalikan dari pewarisan adalah generalisasi.

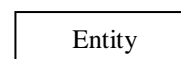
- d. Hubungan dinamis, yaitu rangkaian pesan (*message*) yang di-*passing* dari satu *class* kepada *class* lain.

## 6. ERD

ERD merupakan notasi grafis dalam pemodelan data yang mendeskripsikan hubungan antara penyimpanan. ERD digunakan untuk memodelkan struktur data dan hubungan antar data. Istilah-istilah penting dalam ERD :

### a. Entity

Adalah suatu objek yang dapat diidentifikasi dalam lingkungan pemakai, sesuatu yang penting bagi pemakai dalam konteks sistem yang akan dibuat. Sebagai contoh pelanggan, pekerja dan lain-lain. Seandainya A adalah seorang pekerja maka A adalah isi dari pekerja. Karena itu harus dibedakan antara entiti sebagai bentuk umum dari deskripsi tertentu dan isi entiti seperti A dan B dalam contoh di atas. *Entity* digambarkan dalam bentuk persegi empat. Contoh *Entity* dapat dilihat pada Gambar 1.

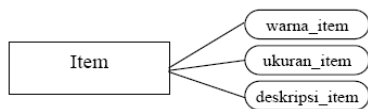


Gambar 1 *Entity*

### b. Atribut

*Entity* mempunyai elemen yang disebut atribut, dan berfungsi mendeskripsikan karakter *entity*. Misalnya atribut *kode\_barang* dari *entity* barang. Setiap *entity* dapat memiliki lebih dari satu atribut. Atribut digambarkan dalam bentuk

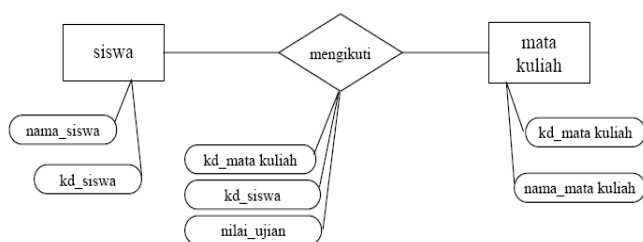
*ellips*. Contoh atribut dapat dilihat pada Gambar 2.



Gambar 2 Atribut

### c. Hubungan

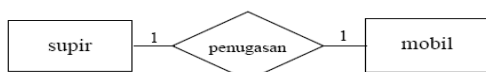
Dalam hubungan harus dibedakan antara bentuk hubungan antar *entity* dengan isi dari hubungan itu sendiri. Misalnya dalam kasus hubungan antara *entity* siswa dan *entity* mata\_kuliah adalah mengikuti, sedangkan isi hubungannya dapat berupa nilai\_ujian. Hubungan digambarkan dalam bentuk intan atau *diamonds*. Contoh hubungan dalam ERD dapat dilihat pada Gambar 3.



Gambar 3 Contoh hubungan dalam ERD

Jenis-jenis hubungan dalam ERD:

- 1) Satu ke satu (*one to one*), misalnya dalam suatu perusahaan mempunyai aturan satu supir hanya boleh menangani satu kendaraan karena alasan tertentu. Hubungan *one to one* dapat dilihat pada Gambar 4.



Gambar 4 Contoh hubungan *one to one*

- 2) Satu ke banyak (*one to many*), misalnya suatu sekolah selalu mempunyai asumsi bahwa satu kelas terdiri dari banyak siswa tetapi tidak sebaliknya, yaitu satu siswa tidak dapat belajar pada kelas yang berbeda. Hubungan *one to many* dapat dilihat pada Gambar 5.



Gambar 5 Contoh hubungan *one to many*

- 3) Banyak ke banyak (*many to many*), misalnya suatu sekolah mempunyai asumsi bahwa siswa mendapatkan banyak mata pelajaran sedangkan mata pelajaran diikuti oleh banyak siswa. Hubungan *many to many* dapat dilihat pada Gambar 6.



Gambar 6 Contoh hubungan *many to many*

Aturan dalam ERD:

- 1) Memilih kunci *record* : dari daftar kamus data, memilih *field* yang akan digunakan sebagai kunci *record*.
- 2) Membentuk *entity* berdasarkan kunci *record*.
- 3) Mencari relasi antar tabel dan menggabungkannya.

## 7. Rancangan dan Implementasi

### 7.1 Analisis Sistem

Pada tahap analisis sistem dilakukan pengumpulan data-data yang diperlukan

dalam pembuatan aplikasi pembayaran sekolah. Metode pengumpulan data yang telah dilakukan adalah sebagai berikut:

- a. Wawancara, yaitu melakukan tanya jawab secara langsung dengan pihak-pihak yang bersangkutan.
- b. Observasi, yaitu melakukan pengamatan dan melihat catatan yang dibutuhkan.

Dokumentasi cara kerja sistem pembayaran manual yang berada di SMA Yos Sudarso Cilacap perlu dilakukan dalam tahapan ini untuk mengidentifikasi alur proses dan aliran data yang diperlukan. Berdasarkan hasil wawancara dan observasi didapatkan alur proses pembayaran yang berjalan di SMA Yos Sudarso Cilacap adalah sebagai berikut:

- a. Proses pembayaran siswa meliputi proses pembayaran SPP, uang gedung dan uang ujian.
- b. Proses pembayaran SPP dari siswa dimulai dengan penyerahan kartu SPP oleh siswa, kemudian pencatatan pembayaran dari siswa oleh petugas administrasi ke dalam buku besar. Setelah itu, petugas menyerahkan bukti pembayaran kepada siswa.
- c. Proses pencarian data pembayaran siswa dilakukan dengan cara mencari data pembayaran siswa per kelas satu per satu pada buku besar sesuai dengan yang dibutuhkan.
- d. Proses pelaporan data pembayaran siswa dilakukan dengan mengumpulkan semua data

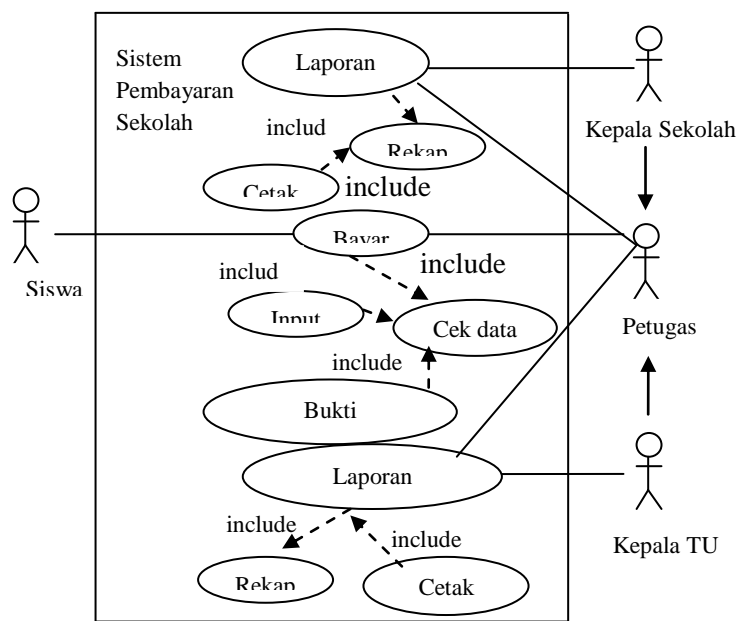
pembayaran siswa tiap kelas yang berada dalam buku besar setiap bulan, semester, dan tahun.

Perangkat keras yang akan digunakan sebagai implementasi sistem pembayaran terkomputasi adalah perangkat keras yang sudah ada di instansi SMA Yos Sudarso Cilacap. Spesifikasi perangkat keras yang berada pada SMA Yos Sudarso Cilacap adalah tiga buah PC (*Personal Computer*) dengan:

- a. *Processor* Intel Pentium 4
- b. RAM (*Random Acces Memory*) 256 Mb
- c. *Hardisk* 40 Gb

## 7.2 Perancangan Sistem

Tahap perancangan sistem ini menggunakan *use case diagram* dan *class diagram*. Use case diagram dapat dilihat pada Gambar 7.



Gambar 7 Use Case Diagram

### 7.3 Perancangan Basis Data

#### 7.3.1 ERD

- a. Langkah 1 (menentukan *entity*)

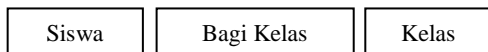
Siswa, bagi kelas, kelas

Siswa, SPP

Siswa, Uang gedung

Siswa, Uang ujian

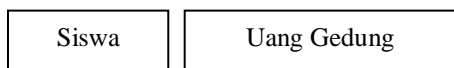
- b. Langkah 2 (menggambarkan *entity*)



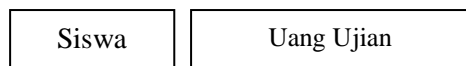
Gambar 8 *Entity* Siswa dengan Bagi kelas



Gambar 9 *Entity* Siswa dengan SPP



Gambar 10 *Entity* Siswa dengan Uang Gedung

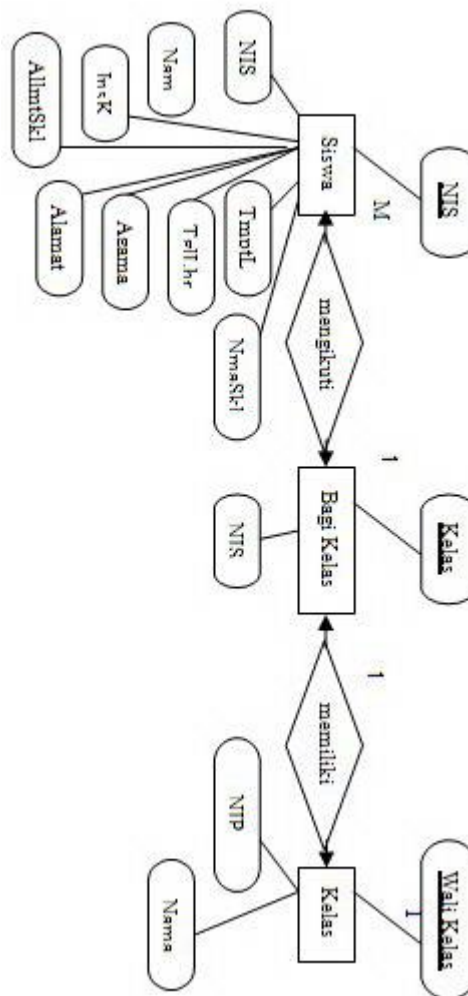


Gambar 11 *Entity* Siswa dengan uang Ujian

- c. Langkah 3 (menggambar relasi)

- d. Melengkapi dengan atribut kunci dan atribut lainnya

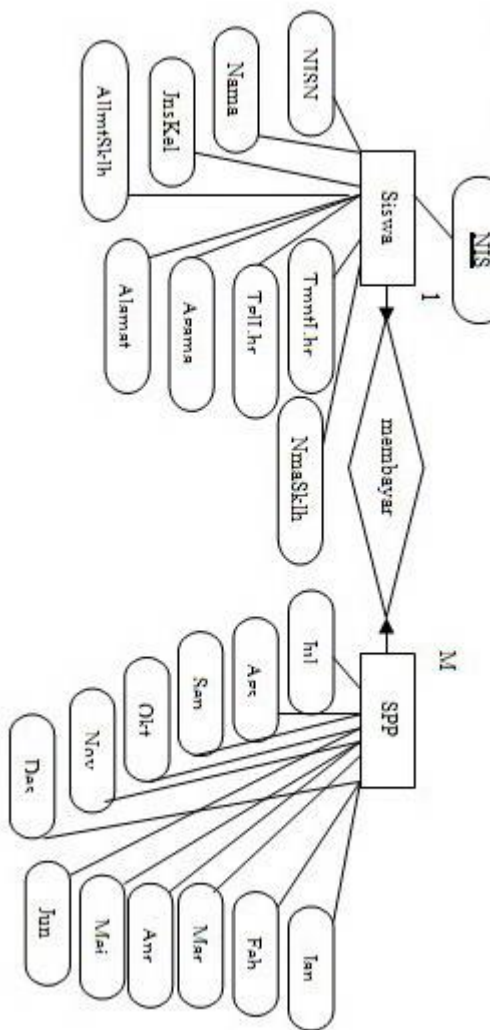
ERD siswa dengan bagi kelas dapat dilihat pada Gambar 12.



Gambar 12 ERD Siswa dengan Bagi Kelas

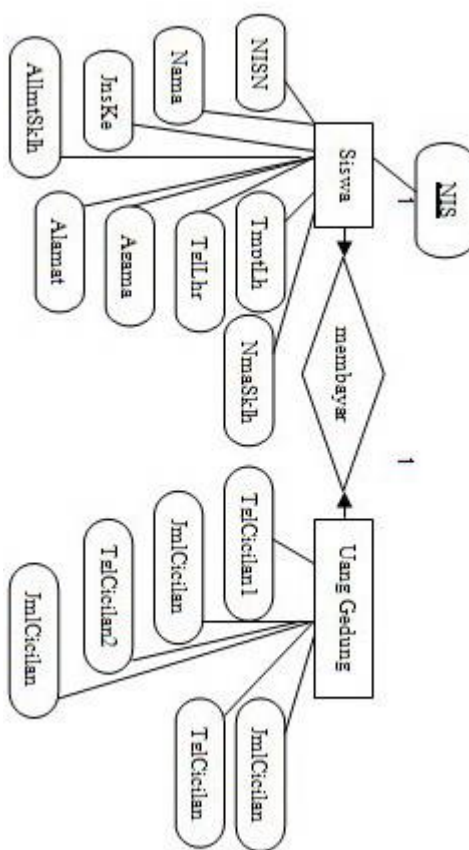
ERD siswa dengan uang SPP dapat dilihat pada Gambar 13.





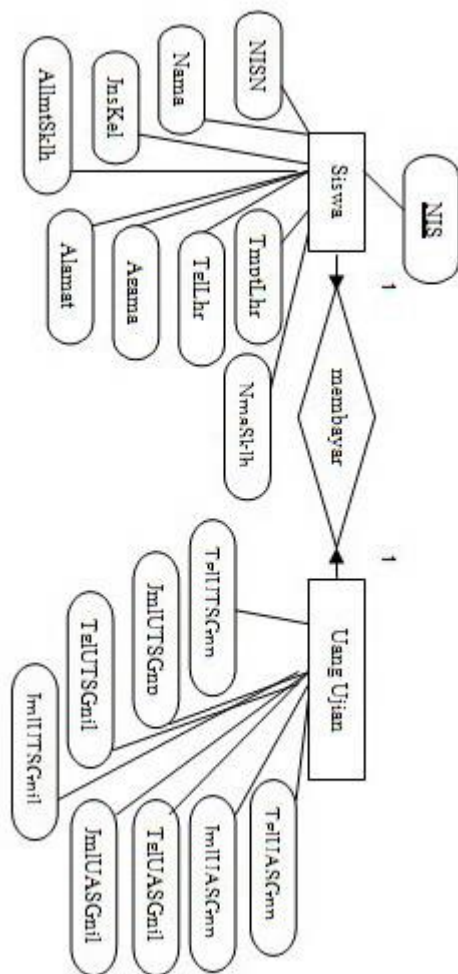
Gambar 13 ERD Siswa dengan Uang SPP

ERD siswa dengan uang gedung dapat dilihat pada Gambar 14.



Gambar 14 ERD Siswa dengan Uang Gedung

ERD siswa dengan uang ujian dapat dilihat pada Gambar 15.



Gambar 15 ERD Siswa dengan Uang Ujian

**7.3.2 Pembuatan Tabel**

Berdasarkan ERD di atas, maka diperoleh tabel-tabel sebagai berikut :

Tabel 1 Tabel Kelas

Nama Field	Tipe	Size
kelas	varchar	15
Wali_kelas	varchar	45
Nip_wali	varchar	20

Tabel 2 Tabel Login

Nama Field	Tipe	Size
username	varchar	20
password	varchar	30
status	varchar	15

Tabel 3 Tabel Uang Gedung

Nama Field	Tipe	Size
nis	varchar	15
tglcicilan1	date	-
ncicilan1	varchar	11
tglcicilan2	date	-
ncicilan2	varchar	11
tglcicilan3	date	-
ncicilan3	varchar	11

Tabel 4 Tabel Bagi Kelas

Nama Field	Tipe	Size
nis	varchar	15
kelas	varchar	8

Tabel 5 Tabel Tahun Ajaran

Nama Field	Tipe	Size
thnAjaran	integer	4

Tabel 6 Tabel SPP

Nama Field	Tipe	Size
nis	varchar	15
juli	date	-
agustus	date	-
september	date	-
oktober	date	-
november	date	-
desember	date	-
januari	date	-
februari	date	-
maret	date	-
april	date	-
mei	date	-
juni	date	-

Tabel 7 Tabel Data Siswa

Nama Field	Tipe	Size
nis	varchar	7
nispn	varchar	12
nama	varchar	45
jenisKelamin	char	1
tptLahir	varchar	45
tglLahir	date	-
agama	varchar	10
alamat	varchar	80

nmSekAsal	varchar	45
almtSekAsal	varchar	45
thnIjazah	varchar	4
noIjazah	varchar	45
tingkat	varchar	10
tglTerima	date	-
nmAyah	varchar	45
nmIbu	varchar	45
kerjaAyah	varchar	20
kerjaIbu	varchar	20
almtOrtu	varchar	80

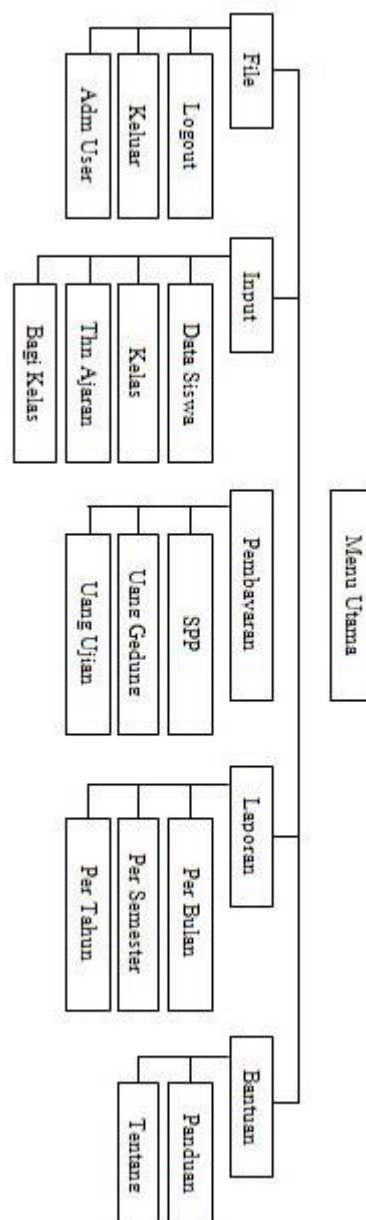
Tabel 8 Tabel Bayar Ujian

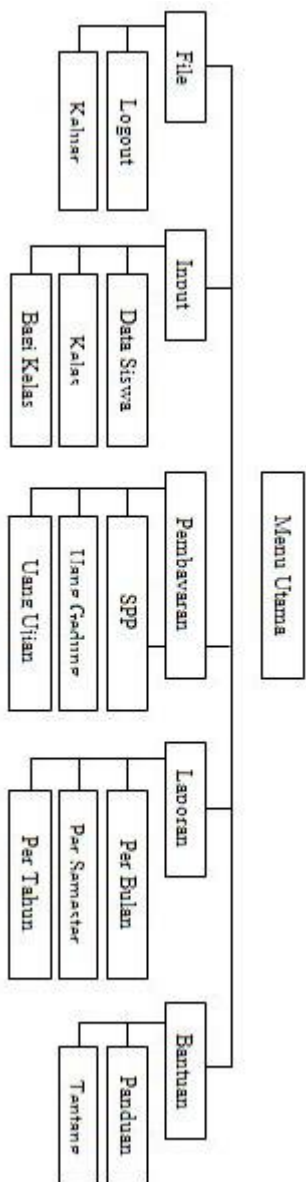
Nama <i>Field</i>	Tipe	Size
nis	varchar	15
tglutsgenap	date	-
jmlutsgenap	varchar	11
tglutsganjil	date	-
jmlutsganjil	varchar	11
tgluasgenap	date	-
jmluasgenap	varchar	11
tglutsganjil	date	-
jmluasganjil	varchar	11

Tabel yang dihasilkan tidak memperlihatkan masih adanya redundansi oleh karena itu dalam penelitian ini tidak dilakukan normalisasi basis data.

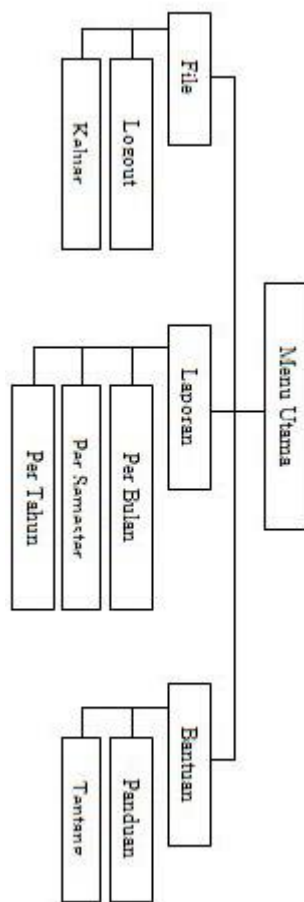
#### 7.4 Perancangan Tingkatan Hak Akses Pengguna

Hak akses pengguna dibagi menjadi tiga tingkatan yaitu head admin, admin dan pengguna biasa. Head admin adalah hak akses untuk kepala tata usaha, hak akses admin dipergunakan untuk petugas di tata usaha sedangkan pengguna biasa dipergunakan oleh kepala sekolah. Perbedaan akses menu masing-masing tingkatan dapat dilihat di Gambar 16, Gambar 17 dan Gambar 18.

Gambar 16 Hak akses untuk *head admin*



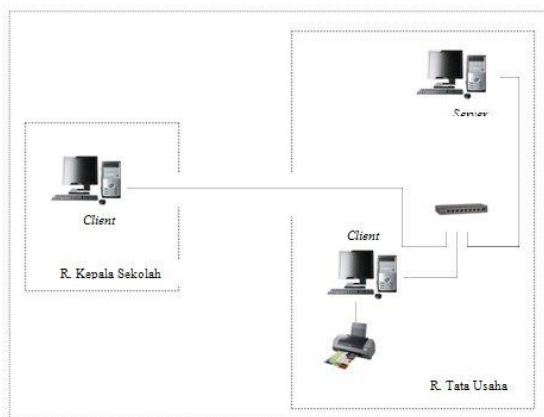
Gambar 17 Hak akses untuk admin



Gambar 18 Hak akses untuk pengguna biasa

### 7.5 Pengaturan Jaringan

Topologi jaringan yang dipergunakan untuk mengimplementasikan aplikasi sistem pembayaran di SMA Yos Sudarso adalah topologi *star*.



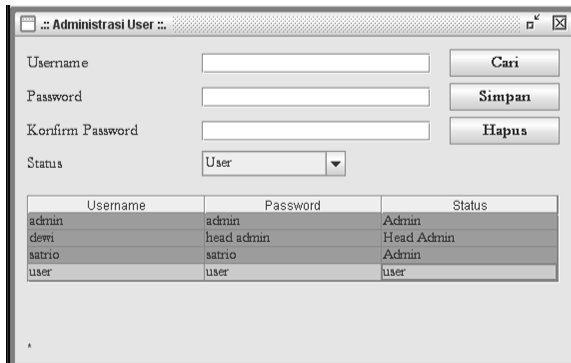
Gambar 19 Topologi jaringan yang diterapkan

### 8. Pengujian

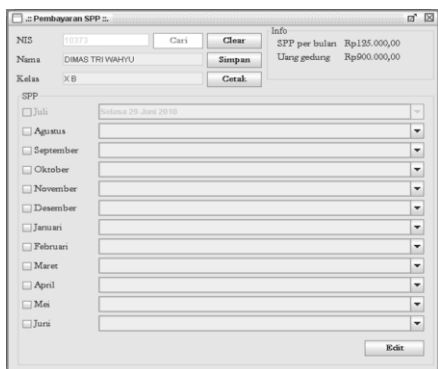
Pengujian dilakukan menggunakan metode *black box testing* dimana pengujian hanya untuk menguji apakah fungsi-fungsi yang ada di dalam aplikasi dapat berjalan dengan semestinya tanpa mempedulikan proses internal dalam kode (Galim, 2004). Aplikasi dijalankan secara *client server* kemudian menu untuk setiap hak akses dijalankan untuk mengetahui apakah semua menu sudah dapat berfungsi. Beberapa tampilan menu dapat dilihat pada Gambar berikut.



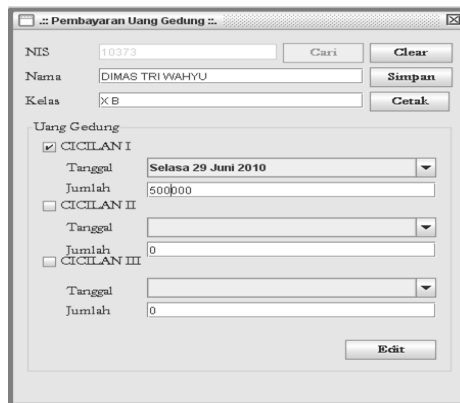
Gambar 20 Menu login



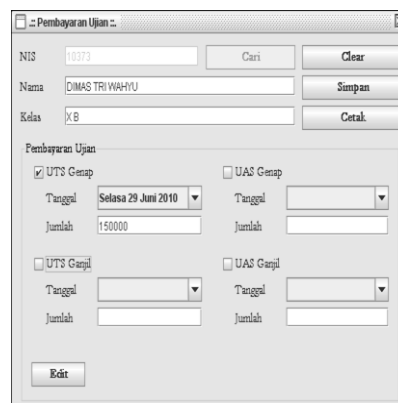
Gambar 21 Menu administrasi user



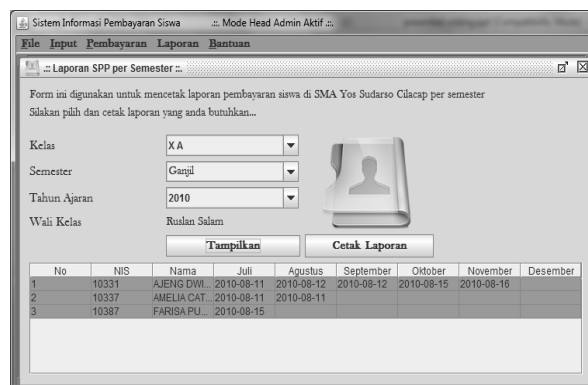
Gambar 22 Menu Pembayaran SPP



Gambar 23 Menu Pembayaran Uang Gedung



Gambar 24 Menu Pembayaran Ujian



Gambar 25 Menu Laporan SPP per semester

### 7.4 Kesimpulan

Dari hasil pengujian, aplikasi sistem pembayaran telah dapat berjalan sesuai dengan fungsi-fungsi yang direncanakan. Aplikasi pembayaran sekolah yang telah dibuat masih memiliki kekurangan yaitu belum adanya kemampuan *backup* data

secara otomatis, tetapi masalah ini dapat diatasi dengan menggunakan aplikasi lainnya. Untuk pengembangan selanjutnya diharapkan kekurangan ini dapat dilengkapi, sehingga aplikasi ini dapat mem-*backup* data secara otomatis.

**Daftar Pustaka:**

Galín, Daniel. 2004. *Software Quality Assurance*. Pearson Education, L.td.

Hariyanto, Bambang. 2007. *Esensi-esensi Bahasa Pemrograman JAVA Edisi 2*. Bandung. Penerbit Informatika.

Kadir, Abdul. 2008. *Tuntunan Praktis Belajar Database Menggunakan MySQL*. Yogyakarta. Penerbit Andi.

Sommerville. 2009. *Software Engineering: Eight Edition*. Pearson Education.