

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang Masalah**

Manajemen keuangan menjadi salah satu faktor yang sangat penting bagi sebuah organisasi manajemen keuangan yang rapi dan memiliki dokumentasi yang lengkap merupakan syarat mutlak terciptanya pelaporan keuangan yang transparan dan akuntabel. Hal tersebut sangat diperlukan untuk menjaga kredibilitas serta nama baik organisasi dimata masyarakat apalagi jika organisasi tersebut merupakan organisasi yang fungsinya sangat strategis dalam kehidupan kemasyarakatan seperti instansi pemerintahan, lembaga sosial masyarakat, perusahaan publik, lembaga pendidikan.

SMK Negeri 2 Temanggung sebagai salah satu lembaga pendidikan yang memiliki peranan yang cukup signifikan dalam dunia pendidikan kejuruan terutama di wilayah kabupaten Temanggung dengan jumlah siswa mencapai lebih dari 1000 orang telah berusaha semaksimal mungkin untuk melakukan manajemen keuangan dengan rapi dan terstruktur dengan baik. Akan tetapi mekanisme pengelolaan keuangan sudah berjalan masih menggunakan cara tradisional yang sangat bergantung pada tenaga, belum ada sistem terkomputasi yang membantu dalam pengelolaan data-data keuangan tersebut. Hal ini tentunya berdampak pada tingkat *human error* yang cukup tinggi sehingga menyebabkan rekapitulasi dan pelaporan keuangan yang dihasilkan kurang akuntabel.

Sistem yang masih belum terkomputasi tersebut juga menyebabkan sangat terbatasnya akses publik terhadap pelaporan keuangan yang update. Stakeholder yang terlibat dalam penyelenggaraan pendidikan di institusi tersebut hanya menerima pelaporan keuangan setiap satu tahun sekali itupun hanya pelaporan yang sangat general. Hal tersebut tentunya juga memperburuk tingkat transparansi pelaporan keuangan yang ada. Dan itu bukan disebabkan karena ketidakmauan pihak yang berwenang untuk melaporkannya secara detil dan lengkap, akan tetapi karena keterbatasan sumber daya teknologi yang bisa menangani hal tersebut.

Solusi yang tepat untuk permasalahan tersebut adalah dengan membangun sebuah sistem terkomputasi untuk mempermudah dan mempercepat proses pencatatan dan pengolahan transaksi keuangan yang terjadi.

## **1.2 Rumusan Masalah**

Lingkup permasalahan dalam laporan kerja praktek ini adalah sebagai berikut :

1. Bagaimana merancang sebuah sistem informasi pengelolaan keuangan berbasis web dengan metode berorientasi objek.
2. Bagaimana mengimplementasikan rancangan sistem informasi tersebut kedalam arsitektur MVC (*Model – View – Controler*).

### **1.3 Batasan Masalah**

Lingkup materi kerja praktek yang dilaksanakan di SMK Negeri 2 Temanggung adalah pembuatan sistem informasi berbasis web yang memiliki fungsionalitas untuk melakukan pengelolaan transaksi keuangan berupa pemasukan, pengeluaran, rekapitulasi serta pelaporannya yang terjadi sehari-hari yang berada dibawah kewenangan petugas keuangan harian. Adapun kewenangan petugas keuangan harian di SMK N 2 temanggung adalah melakukan pencatatan dana iuran siswa, penyetoran dana yang iuran ke bendahara bidang, serta transaksi peminjaman dana skala kecil. Sebagai pendukung fungsi pengelolaan, sistem yang dibangun juga memiliki fasilitas pengelolaan data referensi yaitu : Data Siswa, dan Data Jenis Iuran.

Selain pengelolaan data-data yang telah disebutkan diatas sistem yang dibangun juga memiliki fitur untuk melihat daftar siswa yang terkena black list atau larangan mengikuti ujian serta pencetakan kartu ujian berdasarkan rekapitulasi keuangan siswa yang bersangkutan.

### **1.4 Tujuan**

1. Merancang aplikasi perangkat lunak dengan metode berorientasi objek untuk pengelolaan transaksi keuangan.
2. Mengimplementasikan hasil rancangan dalam bentuk aplikasi berbasis web dengan pendekatan atau arsitektur MVC (*Model-View-Controler*)

## **1.5 Manfaat**

1. Mempermudah pengelolaan dan pelaporan keuangan di SMK N 2 Temanggung
2. Mengaplikasikan ilmu yang diperoleh selama perkuliahan untuk menyelesaikan permasalahan di dunia nyata.
3. Memperoleh tambahan ilmu yang tidak diperoleh selama proses perkuliahan.

## **BAB II**

### **TEMPAT KERJA PRAKTEK**

#### **2.1 Riwayat SMK Negeri 2 Temanggung**

Pendirian SMK Negeri 2 Temanggung dilatarbelakangi oleh berbagai hal, beberapa diantaranya adalah :

1. Pola pembangunan Propinsi Jawa Tengah, bahwa sampai dengan akhir tahun 1964 di setiap Kabupaten di Propinsi Jawa Tengah harus ada minimal 1 SLTA Kejuruan.
2. Sangat dibutuhkannya Sekolah Kejuruan guna memenuhi kehendak masyarakat Kabupaten Temanggung.
3. Hasrat pelajar lulusan SLTP yang ingin melanjutkan ke SLTA Kejuruan.

Maka pada tahun 1968 dibentuklah Tim Pendiri Sekolah Kejuruan dan disetujui oleh pemerintah Kabupaten Temanggung sehingga berdirilah Sekolah Kejuruan di Kabupaten Temanggung sebagai berikut :

Nama Sekolah : SMEA Persiapan Kabupaten Temanggung.

Alamat Sekolah : Kranggan (numpang di SMEP Kranggan Temanggung).

Tahun pembukaan : 1 Januari 1969.

Penanggungjawab : Bupati KDH Temanggung.

Nama Kepala Sekolah : S. Soediro.

Jumlah siswa : 46 siswa.

Jumlah guru : 19 orang.

Waktu mausk : Jam 13.00 - 17.30.

Pada tahun 1971 SMEP Negeri Kranggan pindah lokasi ke Jalan Pahlawan No. 4 Temanggung (sekarang menjadi SMP Negeri 3 Temanggung). Dengan demikian SMEA juga ikut pindah lokasi, namun karena dinilai sudah siap dalam segala sesuatunya, maka nama sekolah diganti menjadi SMEA Kabupaten Temanggung.

Pada tahun 1972 sekolah telah terdaftar dengan No. Urut : 37.A. Dasar Keputusan No: KPPE/015/C.14/III/B.72. tg. 01-01-72. Atas dasar Surat surat Bupati KDH Temanggung No. U.B. 1907/34/72 Tgl 12 Sep. 1972, sekolah dipindah ke Jl. Jenderal Sudirman 30 depan KOMRES 975 Temanggung (sekarang menjadi kantor Camat Temanggung), sedangkan untuk waktu pelajaran siang/sore hari dikarenakan pagi hari digunakan oleh SPG Temanggung.

Pada tahun 1977 atas dasar Surat Dispensasi Ka. Kanwil Depdikbud Propinsi Jawa Tengah dengan No: 02019/A/Set/U/77 Tanggal 2 Mei 1977 BP. Pratomo, BA Guru SMEA Negeri Magelang diangkat sebagai Kepala Sekolah menggantikan Bp.S. Soediro. Gedung yang ditempati pada saat itu akan digunakan Pemda untuk Gedung Balai Pemuda dan Kebudayaan, maka Sekolah dipindah ke Gedung yang baru di Jl. Gilingsari No. 2 Temanggung (Sekarang menjadi SMK Swadaya). Untuk lebih meningkatkan kemandirian Sekolah, maka atas biaya

Pemda dan biaya ganti rugi gedung, SMEA Kabupaten Temanggung pada tahun 1981 pindah lagi ke Jl. Dr. Sutomo 36 Temanggung (sekarang kantor Dinas Pendidikan Nasional).

SMEA Kabupaten Temanggung semakin maju dan diminati masyarakat, dengan demikian Bupati Temanggung (Drs. H. Jacob) memberikan rekomendasi untuk menjadikan SMEA menjadi sekolah negeri dengan no: 421.5/75 tanggal 11 Maret 1982 dengan dibarengi pemberian harga ringan untuk ganti rugi tanah di Jl. Kartini 34B seluas 1 Ha dengan harga hanya Rp. 4.000.000,- (empat juta rupiah) itupun diangsur empat kali.

Dengan segala kemampuan yang ada dan usaha yang tidak pernah berhenti, panitia penerangan yang dikoordinir oleh Bp. Drs. Djenal Soetrisno (saat itu masih kepala SMEA Negeri magelang, saat ini kepala SMK YP 17 “VI” Temanggung) selama kurang lebih 3 tahun, akhirnya dengan SK Mendikbud No. 0559/C/1984 tanggal 20 Nopember 1984 SMEA Swasta Kabupaten Temanggung resmi menjadi sekolah Negeri dengan 6 kelas yang menempati gedung baru di Jl. Kartini dengan sebagian masih di Jl. Dr. Sutomo 36 sementara belum ada kepala sekolah yang definitif, maka kepemimpinan diampu oleh Bp. Drs. Djenal Soetrisno selaku Yang Menjalankan Tugas Kepala Sekolah.

Pada awal tahun 1986/1987 Bp. S. Soeparman dari Klaten diangkat menjadi kepala sekolah, namun baru berjalan sekitar 1 tahun beliau meninggal dunia karena sakit dan untuk kepala sekolah diampu oleh Bp. Ir. Theo Soemarsono kepala STM Pembanungan maron sambil menunggu kepala sekolah yang baru.

Tahun 1988 Bp. Drs. Marwanto dari Solo diangkat menjadi kepala sekolah yang ke-4. Beliau memimpin SMEA Negeri Temanggung ini sampai dengan tahun 1991, selanjutnya yang

menggantikan adalah Bp. Sunardi, BA dari Krapyak Semarang selaku kepala sekolah ke-5 sampai dengan tahun 1996. Yang kemudian menggantikan adalah Bp. Drs. Sihadi sebagai kepala sekolah yang ke-6. Pada saat itu juga sekitar tahun 1997 nama SMEA Negeri Temanggung diganti menjadi Sekolah Menengah Kejuruan (SMK) Negeri 2 Temanggung, untuk SMK Negeri 1 Temanggung-nya adalah STM Pembangunan Maron. Selama kepemimpinan Bp. Drs. Sihadi Jurusan yang semula hanya Akuntansi, Administrasi Perkantoran dan Penjualan bertambah lagi dengan jurusan Tata Boga, Tata Busana dan Teknik Komputer dan Jaringan (TKJ), namun sebelum 6 jurusan tersebut dilengkapi dengan fasilitasnya beliau harus mengemban tugas sebagai pengawas SMK, maka selanjutnya segala tanggungjawab untuk meningkatkan kualitas dan kuantitas serta fasilitas yang harus dimiliki SMK Negeri 2 Temanggung mulai tahun pelajaran 2006/2007 menjadi tanggungjawab kepala Sekolah yang baru atau yang ke-7 yaitu Bp. Drs. Hendro Martono yang semula adalah guru SMK Negeri 2 Temanggung ini juga.

Sampai dengan saat riwayat keberadaan SMK Negeri 2 Temanggung dibuat (per 25 Agustus 2008) dengan segala upaya yang gigih kepala sekolah yang diampu oleh Bp. Drs. Hendro Martono telah melengkapi gedung yang semakin hari sangat dibutuhkan untuk kemajuan SMK Negeri 2 Temanggung dimasa yang akan datang.

## **2.2 Visi dan Misi SMK Negeri 2 Temanggung**

### **2.2.1 Visi**

Membentuk wiraswasta mandiri sebagai basis menuju masyarakat berwawasan global.



### 2.2.2 Misi

Menumbuhkan secara intensif daya juang untuk keunggulan kompetitif bagi seluruh warga sekolah dengan memanfaatkan teknologi informasi dalam proses pembelajaran sehingga tercipta proses belajar mengajar yang aktif, kreatif dan menyenangkan.

Meningkatkan pelayanan prima dalam upaya memberdayakan siswa dan masyarakat.

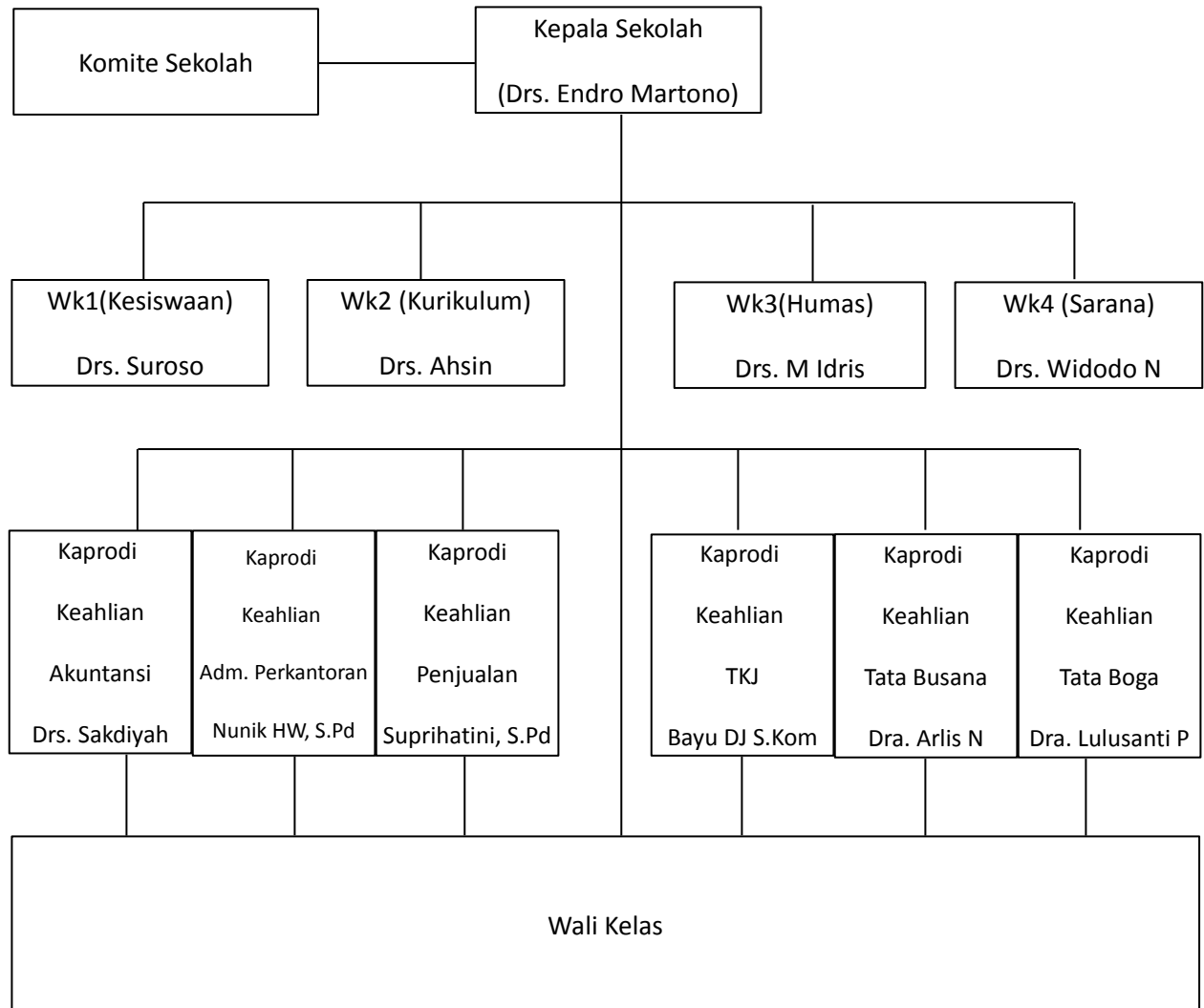
Membangun iklim belajar yang dialogis dan demokratis yang berakar pada norma dan nilai budaya bangsa.

Mengadopsi standar ISO 9001:2000 dengan melibatkan seluruh warga sekolah dan *stake holders*.

Membentuk tamatan yang berkepribadian kuat, berkemampuan adaptif dan produktif sehingga mampu menghidupi dirinya dengan meminimalkan ketergantungannya terhadap orang lain.

## 2.3 Struktur Organisasi SMK N 2 Temanggung

Dibawah ini merupakan gambar bagan struktur organisasi di SMK Negeri 2 Temanggung pada saat dilangsungkannya kerja praktek.



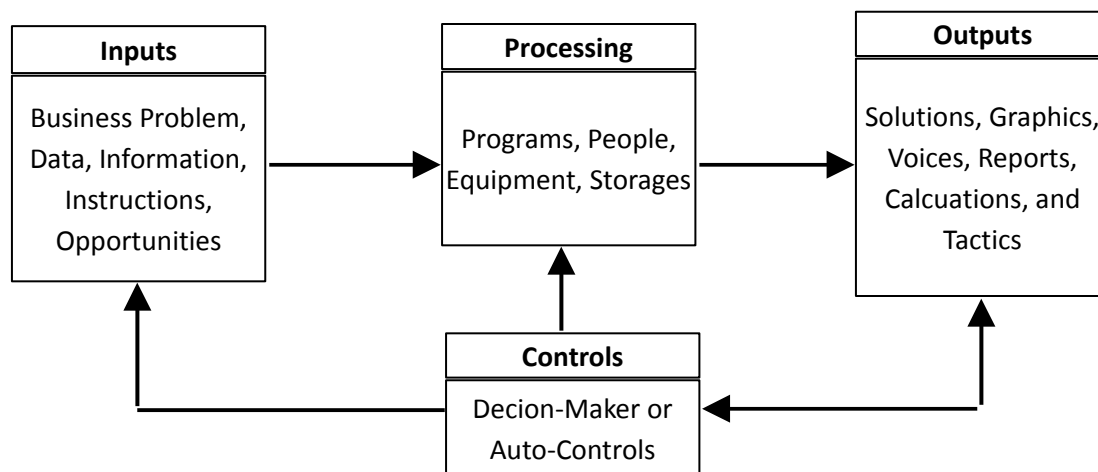
Gambar 2.1 Struktur Organisasi SMK N 2 Temanggung

## BAB III

### LANDASAN TEORI

#### 3.1 Konsep Sistem Informasi

Suatu sistem informasi dapat didefinisikan secara teknis sebagai satuan komponen yang saling berhubungan yang mengumpulkan (atau mendapatkan kembali), memproses, menyimpan, dan mendistribusikan informasi untuk mendukung pengambilan keputusan dan kendali dalam suatu organisasi dan juga membantu stakeholder untuk meneliti permasalahan, memvisualisasikan pokok-pokok yang kompleks, dan menciptakan produk-produk baru (Laudon & Laudon, 2003). Dalam definisi lain istilah system informasi bisa diartikan sebagai sebuah kombinasi dari perangkat keras (*hardware*), perangkat lunak (*software*), dan jaringan telekomunikasi yang dibangun untuk mengumpulkan, mengolah data dan mendistribusikannya dalam bentuk informasi yang berguna dalam suatu organisasi (Jessup & Valacich, 2003)



Gambar 3.1 Skema Sistem Informasi (Jessup & Valacich, 2003)

Sistem Informasi berisi informasi tentang unsur-unsur yang ada dalam sebuah organisasi. Informasi berarti data yang telah dibentuk ke dalam suatu format yang mempunyai arti dan berguna bagi manusia. Sebaliknya data merupakan sekumpulan baris fakta yang mewakili peristiwa yang terjadi pada organisasi atau pada lingkungan fisik sebelum diolah kedalam suatu format yang dapat dipahami dan digunakan oleh orang (Laudon & Laudon, 2003).

Sistem informasi menerima input berupa data atau instruksi, memproses input sesuai instruksi kemudian mengeluarkan hasil pemrosesan berupa output. Selain itu dalam sistem informasi juga diperlukan adanya media penyimpanan file (*data file storage*) untuk menyimpan hasil pengolahan sebelumnya sehingga dapat digunakan kembali ketika diperlukan. Sistem informasi terdiri dari beberapa komponen, antara lain :

- a) **Blok Masukan**, Berupa metode-metode dan media untuk menangkap data yang akan dimasukkan yang biasanya berupa dokumen-dokumen dasar.
- b) **Blok Model**, Berupa kombinasi prosedur, logika dan model matematik yang berfungsi memanipulasi data untuk keluaran tertentu.
- c) **Blok Output**, Berupa data-data keluaran misalnya informasi atau laporan-laporan tertentu.
- d) **Blok Teknologi**, Berupa teknologi yang digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan output serta membantu pengendalian sistem secara keseluruhan. Blok teknologi ini merupakan komponen yang membantu memperlancar proses pengolahan data di dalam sistem.

- e) **Blok Basis Data**, Berupa kumpulan data yang saling berhubungan yang tersimpan di perangkat keras dan perangkat lunak komputer.
- f) **Blok Kendali**, Berupa mekanisme yang berfungsi untuk mencegah dan menangani kesalahan/kegagalan dalam sistem.

Seiring kemajuan zaman, perkembangan sistem informasi semakin pesat. Hal ini karena manfaat sistem informasi yang sangat besar dalam membantu aktivitas organisasi. Sekarang hampir semua jenis organisasi berlomba-lomba untuk menerapkan sistem informasi dalam setiap aktivitasnya.

### **3.2 Konsep Sistem Informasi Manajemen Keuangan**

Menurut terminologinya, Sistem Informasi Manajemen Keuangan biasanya merujuk pada proses komputerisasi manajemen proses pembelanjaan publik termasuk didalamnya adalah formulasi anggaran, eksekusi anggaran, dan pelaporan (Robin-Brown, 2008).

Definisi lain menjelaskan bahwa Sistem Informasi Manajemen Keuangan merupakan sistem informasi yang membuat pelacakan mengenai transaksi finansial untuk kemudian dijadikan informasi yang bisa dipahami. Sistem ini juga mendukung pelaporan manajerial, pembuatan kebijakan, dan laporan pertanggungjawaban (Dorotinsky, 2003).

International Monetary Fund (IMF) telah menetapkan standarisasi untuk sistem informasi manajemen keuangan internasional seperti dibawah ini (Diamond & Khemani, 2005) :

- a) Modular dan mampu diperbaharui sesuai dengan kebutuhan di masa depan

- b) Menawarkan platform yang general dan antar muka yang mampu mengakomodasi stakeholder dari berbagai macam disiplin dan dapat dipertanggungjawabkan dalam proses manajemen keuangan dalam menambah dan mengakses basis data yang tersedia.
- c) Mampu memelihara basis data historikal dari rancangan anggaran dan pengeluaran, data transaksi dalam tingkat kedetailan yang tinggi.
- d) Memiliki modul yang mampu menangani estimasi pendapatan dan pengeluaran jangka pendek, jangka menengah maupun jangka panjang.
- e) Memiliki perangkat *build in* yang mampu melakukan analisis terhadap trend dari berbagai macam elemen operasi fiskal.
- f) Cukup fleksibel untuk menyediakan dan mengelola informasi yang *user-defined*.

### **3.3 Konsep Basis Data dan DBMS (Database Management Sistem)**

Basis data adalah sekumpulan informasi terutama informasi yang saling berhubungan dan terorganisir (Powell, 2006). Database merupakan salah satu komponen yang penting dalam sistem informasi, karena berisi basis penyedia informasi bagi para pemakainya. Penerapan database dalam sistem informasi disebut dengan database sistem. Sistem basis data ( database system ) adalah suatu sistem informasi yang mengintegrasikan kumpulan dari data yang saling berhubungan satu dengan yang lainnya dan yang membuatnya tersedia untuk beberapa aplikasi yang bermacam macam dalam suatu organisasi (Jogiyanto, 2000).

Secara teknis basis data adalah kumpulan dari beberapa unsur dibawah ini :

**a. Tabel**

Tabel merupakan inti dari sebuah basis data, untuk menyimpan data yang dikelompokkan dalam bentuk baris dan kolom, setiap baris mewakili record dan setiap kolom mewakili field (atribut).

**b. View**

Merupakan tabel maya yang isinya ditentukan oleh query ke dalam basis data.

**c. Indeks**

Indeks adalah file jenis khusus yang bekerja sama dengan tabel, bertujuan mempercepat proses pengaksesan record atau sekelompok record tertentu.

**d. Trigger**

Adalah Prosedur tersimpan yang secara otomatis dijalankan apabila data data di dalam tabel berubah karena eksekusi perintah SQL (*Structured Query Language*) yang meliputi insert, update atau delete.

**e. Prosedur**

Prosedur adalah pengaksesan data di dalam basis data melalui eksekusi perintah SQL.

Bahasa standar untuk melakukan operasi atau query terhadap basis data adalah SQL (*Structure Query Language*). Didalam SQL dikenal tiga sub bahasa yang digunakan dalam

pengolahan basis data yaitu DDL (*Data Definition Leanguage*), DML (*Data Manipulation Leanguage*), dan DCL (*Data Control Leanguage*).

a) **DDL (*Data Definition Leanguage*)**

DDL digunakan dalam membuat struktur basis data mulai dari pendefinisian basis data, table, view, index, serta perintah yang berhubungan dengan maintenance dan struktur basis data tersebut. Contoh query SQL yang termasuk kedalam golongan DDL adalah : Create, alter, drop.

b) **DML (*Data Manipulation Leanguage*)**

DML digunakan untuk memanipulasi data dalam basis data seperti menambah, mencari, mengubah, ataupun menghapus data dari table. Contoh query SQL yang termasuk kedalam DML antara lain : insert, select, delete, update

c) **DCL (*Data Control Leanguage*)**

Merupakan perintah atau query yang fungsinya adalah membantu dalam melakukan pengawasan keamanan basis data dengan query ini dimungkinkan untuk memberikan otoritas kepada user tertentu untuk mengakses basis data, mengalokasikan ruang dalam basis data, serta pengauditan penggunaan basis data adapun perintah yang termasuk kedalam DCL antara lain adalah *grant* dan *revoke*.

DBMS atau Database Management System adalah perangkat lunak yang didesain untuk membantu dan memanfaatkan kumpulan data yang besar (Ramakkrishnan & Gehrke, 2003). Penggunaan DBMS memiliki beberapa manfaat penting diantaranya adalah :



1. **Kemandirian Data** : Program aplikasi idelanya tidak diekspos pada detail representasi dan penyimpanan data. DBMS menyediakan satu pandangan abstrak tentang data yang menyembunyikan detail tersebut
2. **Akses Data Efisien** : DBMS memanfaatkan berbagai teknik yang canggih untuk menyimpan dan mengambil data secara efisien. Fitur ini terutama penting jika data disimpan pada alat penyimpanan eksternal
3. **Integritas dan Keamanan Data** : Jika data selalu diakses melalui DBMS, maka DBMS dapat memanfaatkan batasan integritas.
4. **Administrasi Data** : Ketika beberapa pengguna berbagi data, pemusatan administrasi data dapat memberikan perbaikan yang signifikan pada penggunaan data. Representasi data dapat dengan mudah diatur sedemikian rupa sehingga dapat meminimalisir redundansi dan untuk mem-finetune penyimpanan data guna melakukan pengambilan data yang efisien.
5. **Akses Konkuren dan *Crash Recovery*** : DBMS menjadwalkan akses konkuren pada data dalam cara tertentu sehingga pengguna dapat memandang data sebagai data yang sedang diakses oleh hanya satu pengguna pada satu waktu. Lebih lanjut DBMS memproteksi pengguna dari efek kegagalan system.
6. **Waktu Pengembangan Aplikasi berkurang** : DBMS mendukung fungsi penting yang merupakan hal biasa bagi banyak aplikasi untuk mengakses data dalam DBMS. Hal ini, dalam kaitannya dengan interface level tinggi pada data, memfasilitasi pengembangan

aplikasi yang cepat. Aplikasi DMS kemungkinan besar menjadi lebih kuat daripada aplikasi yang berdiri sendiri karena banyak tugas penting bisa ditangani oleh DBMS.

Beberapa sistem basis data yang menggunakan SQL antara lain: Oracle, MS SQL Server, PostgreSQL, MySQL, Firebird dan MS Access. Perintah SQL dibedakan dalam tiga hal yaitu DDL (Data Definition Language), DML (Data Manipulation Language) dan DCL (Data Control Language).

### **3.3.1. Database Server dengan MySQL.**

MySQL dikembangkan oleh sebuah perusahaan Swedia bernama MySQL AB, yang kala itu bernama TcX DataKonsult AB, sejak sekitar 1994–1995, meski cikal bakal kodenya bisa disebut sudah ada sejak 1979. Tujuan mula-mula TcX membuat MySQL pada waktu itu juga memang untuk mengembangkan aplikasi Web untuk klien-TcX adalah perusahaan pengembang software dan konsultan database. Kala itu Michael Widenius, atau “Monty”, pengembang satu-satunya di TcX, memiliki aplikasi UNIREG dan rutin ISAM yang dibuat sendiri dan sedang mencari antarmuka SQL untuk ditempelkan di atasnya. Mula-mula TcX memakai mSQL, atau “mini SQL”. Barangkali mSQL adalah satu-satunya kode database open source yang tersedia dan cukup sederhana saat itu, meskipun sudah ada Postgres. Namun ternyata, menurut Monty, mSQL tidaklah cukup cepat maupun fleksibel. Versi pertama mSQL bahkan tidak memiliki indeks. Setelah mencoba menghubungi David Hughes, pembuat mSQL dan ternyata mengetahui bahwa David tengah sibuk mengembangkan versi dua, maka keputusan yang diambil Monty yaitu

membuat sendiri mesin SQL yang antar mukanya mirip dengan mSQL tapi memiliki kemampuan yang lebih sesuai kebutuhan. Lahirlah MySQL.

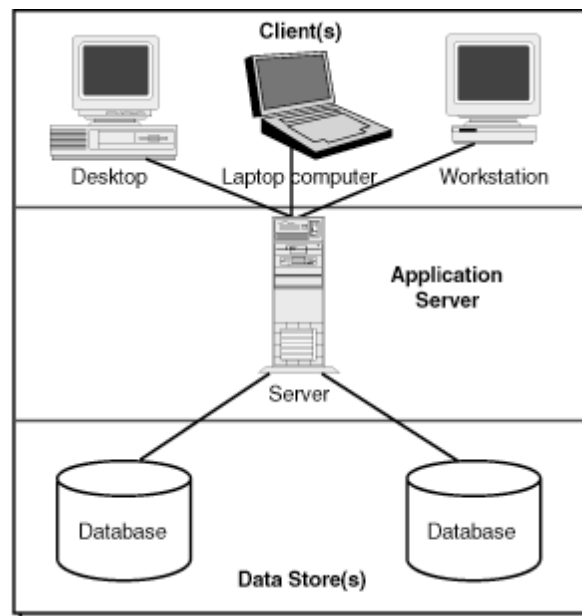
MySQL merupakan software Database Management System yang sangat populer di kalangan pemrograman web, khususnya yang berbasis open source. Hal ini disebabkan karena kemudahannya untuk digunakan. MySQL telah mendapatkan lisensi open source GNU General Public License (GPL) mulai versi 3.23 pada bulan juni 2000, sehingga dapat digunakan oleh siapa saja dan kapan saja tanpa harus membayar biaya lisensi. Pada awalnya mysql hanya dapat dioperasikan pada satu platform saja. Namun dalam perkembangannya, mysql dapat dioperasikan dalam berbagai platform seperti windows, FreeBSD, ataupun linux.

### **3.4 Konsep Aplikasi Berbasis Web**

Aplikasi sistem informasi berbasis web dibangun berdasar model arsitektur tiga layer . Bagian dasar aplikasi ini adalah layer basis data (*data layer*), terdiri dari database management system (DBMS) yang mengatur basis data berupa membuat data pemakai, menghapus, memodifikasi dan query. Di atas layer basis data adalah layer tengah (*middle layer*) yang berisi logika bisnis (*bussines logic*) ataupun aturan dan prosedur pengolahan. Tugas utama layer ini adalah mengkomunikasikan anatara layer basis data dengan layer paling atas yang berhubungan langsung dengan pengguna akhir (*end user*). Layer paling atas ini disebut layer presentasi (*presentation layer*), biasanya web browser yang berperan dalam aplikasi ini.

Web meliputi tiga hal standar yaitu Hypertext Markup Language (HTML), Hypertext Transfer Protocol (HTTP) dan protokol jaringan. Secara singkat HTML merupakan sekumpulan tag yang berfungsi untuk menerjemahkan informasi agar dapat dibaca atau dipahami oleh

manusia melalui browser. Sedangkan HTTP adalah aturan-aturan atau prosedur untuk pengiriman atau pertukaran data antara server (web server) dengan client (browser). Jika HTML merupakan manifestasi dari penerjemahan informasi di sisi client. Maka pengolahan data dan informasi di sisi server bisa direpresentasikan oleh beberapa bahasa pemrograman web seperti PHP, ASP, JSP, CGI-Perl, dlsb. Sedangkan protokol jaringan merupakan kumpulan prosedur untuk mengatur komunikasi antara perangkat keras di sisi server dan perangkat keras di sisi client.



Gambar 3.2 Arsitektur Aplikasi 3-Layer (Cradisky, 2001)

### **3.4.1. Pemrograman Berbasis Web dengan PHP.**

PHP dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada awalnya merupakan skrip-skrip Perl sederhana yang dibuat dengan tujuan untuk mengetahui siapa saja yang telah mengunjungi homepage pribadinya. Skrip-skrip tersebut diberi nama Personal Home Page (PHP). Kemudian ditulis ulang menggunakan bahasa C dan lahir PHP/F1 tahun 1996.

PHP merupakan singkatan dari PHP: Hypertext Preprocessor. PHP adalah bahasa skrip berbasis server (server-side scripting language), seperti ASP, JSP, ColdFusion. Karenanya, skrip PHP dieksekusi di server. PHP mendukung beragam database (MySQL, Informix, Oracle, Sybase, Solid, PostgreSQL, Generic ODBC, dll).

PHP digunakan untuk pembuatan web dinamis. Web dinamis adalah web yang informasinya dapat berubah. Informasi bisa diambil dari database, cookies maupun input user langsung melalui formulir atau URL di web browser. PHP mendukung banyak paket basis data baik berlisensi maupun open source seperti PostgreSQL, mSQL, MySQL, Oracle, Informix, Microsoft SQL Server dan lain-lain. PHP mendukung aplikasi web database skala besar. Ada beberapa alasan yang membuat PHP menjadi pilihan , yaitu:

1. PHP merupakan software yang open source.
2. Script PHP dapat disimpan ke dalam file HTML statis dan ini membuat client tier mudah terintegrasi.
3. Mempunyai kemampuan mengeksekusi secara cepat ke database.
4. Fleksibel dalam platform dan sistem operasi.

### 3.5 Pengembangan Sistem Berorientasi Objek

Pendekatan berorientasi objek pada pengembangan sistem didasarkan pada konsep tentang objek yang telah ada di dalam sebuah lingkungan sistem (Whiten, 2004). Bagian tersulit dari pengembangan perangkat lunak dan/atau sistem informasi adalah tahap analisis di mana kita harus menganalisis masalah yang sangat rumit, yang kita jumpai di dunia nyata serta melakukan abstraksi terhadap masalah itu, kemudian melakukan perancangan agar kelak dapat diimplementasikan dengan cepat serta akurat pada komputer menjadi suatu sistem/perangkat lunak yang sesuai dengan kebutuhan serta harapan pengguna (Nugroho, 2005). Implementasi dari pengembangan sistem berorientasi objek ini adalah menggunakan bahasa pemrograman berorientasi object (*object-oriented programming*).

#### 3.5.1 Kelas dan Objek

Kelas digunakan adalah untuk memodelkan abstraksi yang berasal dari persoalan yang akan diselesaikan(pada proses analisis). Kelas juga digunakan untuk memodelkan abstraksi dari teknologi yang digunakan untuk mengimplementasikan solusi terhadap masalah(hal ini dilakukan pada saat perancangan atau pemrograman). Kelas merupakan struktur statik. Kelas merupakan pengkapsulan nilai-nilai atribut dan operasi-operasi eksklusifnya. Kelas merupakan deskripsi dari suatu objek dengan sekumpulan atribut dan operasi yang seragam.

Kelas merupakan blok pembangun di pendekatan berorientasi objek. Kelas objek mendeskripsikan kelompok objek dengan property-properti(atribut-atribut) serupa, perilaku(operasi-operasi) yang umum, kelas mendeskripsikan sekelompok struktur data yang dapat dibentuk oleh kelas, disebut instan kelas atau objek.

Perbedaan kelas dengan objek adalah objek merupakan entitas kongkret yang ada secara ruang dan waktu sedangkan kelas hanya merupakan representasi abstraksi. Kelas menangkap struktur dan perilaku umum di semua objek instanya. Objek-objek disatu kelas mempunyai atribut-atribut dan pola-pola perilaku serupa. Kebanyakan objek menurunkan individualitasnya dengan perbedaan-perbedaan dalam nilai-nilai atributnya.

Kegunaan kelas :

- a) Mengelompokkan objek-objek menjadi kelas-kelas berarti mengabstraksikan masalah yang sedang dihadapi.
- b) Definisi-definisi umum(seperti nama kelas atau atribut) cukup disimpan sekali dalam suatu kelas, bukan sekali dalam sebuah instan kelas atau(objek). Operasi-operasi dapat ditulis sekali perkelas. Semua objek dalam kelas dapat menggunakan operasi-operasi tersebut berulang-ulang.
- c) Kelas merupakan penerapan prinsip pengkapsulan yang memadukan property-property unit komputasi seperti data dan operasi.

### **3.5.2 Abstraksi**

Abstraksi, seperti yang telah dijelaskan di atas secara sepintas, pada dasarnya adalah menemukan hal – hal yang esensial pada suatu objek dan mengabaikan hal – hal yang sifatnya insidental. Maksud dari pernyataan ini diantaranya adalah menangkap sesuatu yang berarti untuk dituangkan dalam sistem/perangkat lunak alih – alih menangkap seluruh fakta yang ada. Pada pengembangan sistem informasi, ini berarti memfokuskan pada apa itu objek, apa ciri – ciri yang

dimiliki objek tersebut (atribut – atributnya) serta apa yang objek lakukan (operasi – operasi apa yang dilakukan oleh suatu objek) sebelum memutuskan bagaimana ia dapat diimplementasikan.

Banyak bahasa pemrograman modern menyediakan abstraksi namun kemampuan bahasa pemrograman berorientasi objek untuk menggunakan pewarisan dan polimorfisme menyediakan fasilitas tambahan yang sangat diperlukan. Penggunaan konsep abstraksi selama **analisis** berarti *“jangan pernah melakukan perancangan dan implementasi sebelum persoalan benar – benar dipahami.”* Pernyataan ini penting sekali sebab –seperti disinggung di atas– kesalahan saat analisis akan beresiko tinggi menghasilkan sistem dan/atau perangkat lunak yang tidak sesuai dengan apa yang diinginkan oleh calon pengguna.

### **3.5.3 Pembungkusan (*Encapsulation*) dan Pengiriman Pesan (*Message Passing*)**

Pembungkusan (penyembunyian informasi) berarti meninggalkan aspek eksternal dari objek yang dapat dimasup (diakses) oleh objek lain dan memfokuskan diri pada implementasi internal suatu objek. Rincian implementasi internal dari suatu objek *tersembunyi* dari objek – objek lain dan terpisah dari implementasi internal, yaitu antarmuka (*interface*) satu objek dengan objek lainnya. Sebab itu, implementasi internal suatu objek dapat diubah tanpa mempengaruhi aplikasi yang menggunakannya asalkan perubahan internal itu tidak mengubah antarmuka (*interface*) yang digunakan objek yang bersangkutan untuk berkomunikasi dengan objek lainnya.

Pembungkusan sebenarnya tidak unik pada bahasa pemrograman berorientasi objek (Konsep ADT/Abstract Data Type telah digunakan pada bahasa pemrograman terstruktur seperti Pascal serta C) tetapi kemampuan bahasa berorientasi objek untuk menggabungkan struktur data dan perilaku (baca: fungsi atau prosedur) dalam suatu entitas tunggal membuat bahasa



berorientasi objek lebih bardaya-guna dibandingkan bahasa konvensional yang memisahkan struktur data dan perilaku.

Seperti yang kita harapkan, objek adalah dinamis. Objek dapat menerima satu atau lebih metoda untuk menyelesaikan suatu tugas tertentu. Kita dapat merangsang pengaktifan suatu metoda pada objek tertentu dengan cara mengirimkan pesan (*message*). Pesan – pesan harus mengandung nama dari objek yang dituju, nama metoda yang ingin diakses, serta parameter – parameter yang dibutuhkan. Objek menerima pesan dengan parameter – parameter yang dibutuhkan kemudian melakukan suatu metoda tertentu berdasarkan nilai parameter yang diberikan.

Keuntungan dari pembungkusan adalah kita dapat mengharapakan suatu objek melakukan metoda apa yang kita inginkan tanpa harus tahu bagaimana objek itu melakukannya. Kita ibaratkan suatu objek dengan televisi. Kita tidak perlu tahu bagaimana televisi melakukan suatu tugas tertentu, misalnya menayangkan gambar tertentu. Yang perlu kita ketahui adalah tombol mana pada *remote control* yang harus ditekan, kemudian televisi akan berfungsi. Penekanan tombol pada *remote control* mengirimkan pesan tertentu (baca: *message*) pada televisi, memberitahu metoda apa yang akan dilakukan (pindah saluran, mengeraskan suara, meningkatkan intensitas warna tertentu, dan sebagainya). Sekali lagi, pengguna televisi tak perlu tahu bagaimana televisi melakukan semua metoda itu, ia hanya perlu tahu apa yang bisa dilakukan televisi untuknya kemudian mengirimkan pesan (*message*) tertentu agar televisi melakukan hal – hal (baca: metoda) yang dia inginkan.

### 3.5.4 Generalisasi dan Polimorfisme

Teknologi berorientasi menambah 2 mekanisme penting pada tipe data abstrak (ADT/*Abstract Data Type*) yang sepintas telah dijelaskan di atas yaitu: generalisasi/spesialisasi serta polimorfisme.

Generalisasi/spesialisasi memungkinkan kelas – kelas berbagi data serta perilaku yang sama. pada konteks pemrograman, ini memungkinkan pengurangan ukuran kode dan menyediakan kemungkinan pengembangan sistem/perangkat lunak yang lebih mudah dipelihara. Sebagai tambahan, generalisasi juga meningkatkan keterpaduan dalam suatu kelas mengurangi derajat kesaling-bergantungan (meningkatkan koherensi) kelas saat implementasi kelak. Polimorfisme mengijinkan penyesuaian berbagi kode untuk memenuhi keadaan tertentu. Dua mekanisme ini saling bekerjasama untuk mendukung kemandirian objek individual serta mendukung proses pengembangan sistem/perangkat lunak berkelanjutan dimana komponen – komponen yang dibuat pada suatu aplikasi yang memerlukan layanan (*service*) yang sama (Sekali lagi: aplikasi yang terakhir tak perlu tahu rincian implementasi komponen yang digunakannya).

Selain itu, dua mekanisme di atas sangat membantu untuk penorganisasian kelas/objek dan sangat membantu dalam konteks pewarisan (*inheritance*) yang memungkinkan penggunaan ulang komponen (*reuseable components*). Pewarisan ini adalah konsep yang sangat penting dalam teknologi berorientasi objek karena -tentu saja- lebih menghemat biaya dan waktu jika kita bisa menggunakan komponen yang sudah ada untuk sebuah pengembangan sistem/perangkat lunak, alih – alih membuat segala sesuatunya dari awal. Selain itu, fleksibilitas komponen –

komponen pada terminologi berorientasi objek lebih memungkinkan pengembangan untuk menangkap semua kebutuhan serta harapan pengguna (*user's needs and expectations*).

### 3.6 UML (*Unified Modelling Language*)

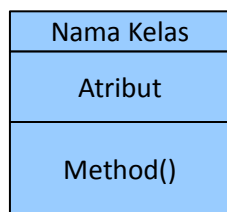
*Unified Modelling Language* (UML) adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka ia 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 modeling aplikasi prosedural dalam VB atau C (Dharwiyanti & Wahono, 2004).

UML memungkinkan *developer* untuk menspesifikasi, memvisualisasi serta mendokumentasikan model sistem dengan cara yang mendukung skalabilitas, keamanan, dan kehandalan eksekusi. Dikarenakan pemodelan UML mampu mencapai level abstraksi melalui tahapan desain dan analisis. Hal tersebut memudahkan dalam pengidentifikasian pola dan behaviour dan hal tersebut memungkinkan untuk *refactoring* dan *reusability*. Konsekuensinya, pemodelan UML bisa memfasilitasi hasil desain yang modular (Pender, 2003).

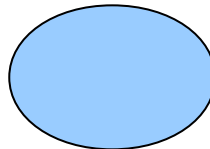
### 3.6.1 Bangunan Dasar UML

Metodologi UML menggunakan 3 bangunan dasar untuk mendeskripsikan sistem/perangkat lunak yang akan dikembangkan yaitu :

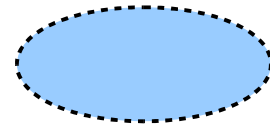
1. Sesuatu (*Things*), ada empat macam *things* dalam UML, yaitu:
  - *Structural Things*, merupakan bagian yang relatif statis dalam model UML. Bagian statis ini dapat berupa elemen-elemen yang bersifat fisik maupun konseptual. Secara keseluruhan ada 7 macam structural things ini yaitu : Kelas, Antar Muka, Kolaborasi, Use Case, Kelas Aktif, Komponen dan Simpul. Notasi untuk masing-masing struktural things dapat digambarkan sebagai berikut :



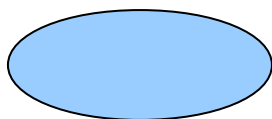
Gambar 3.3 Kelas (Nugroho, 2005)



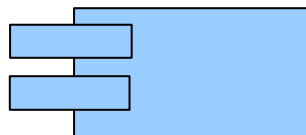
Gambar 3.4 Antarmuka (Nugroho, 2005)



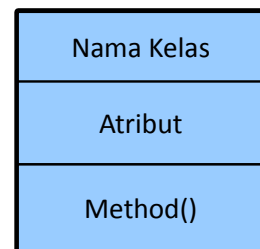
Gambar 3.5 Kolaborasi (Nugroho, 2005)



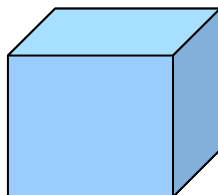
Gambar 3.6 Use Case (Nugroho, 2005)



Gambar 3.7 Komponen (Nugroho, 2005)



Gambar 3.8 Kelas Aktif (Nugroho, 2005)



Gambar 3.9 Simpul (Nugroho, 2005)

- *Behavioral Things*, merupakan bagian yang paling dinamis pada model UML, biasanya merupakan kata kerja dari model UML, yang mencerminkan perilaku sepanjang ruang dan waktu. Ada dua macam behavioral things dalam UML yaitu Interaksi dan State. Notasi UML untuk *behavioral things* adalah sebagai berikut :

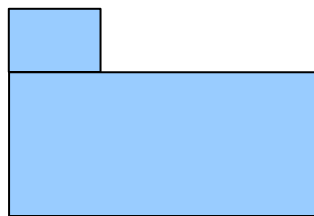


Gambar 3.10 Pesan (Nugroho, 2005)



Gambar 3.11 State (Nugroho, 2005)

- *Grouping Things*, merupakan bagian pengorganisasi dalam UML. Pola pengorganisasiannya adalah dengan penyederhanaan model UML yang digambarkan dalam sebuah paket yang dapat didekomposisi lebih lanjut. Notasi untuk paket digambarkan dalam gambar 3.12 dibawah ini :

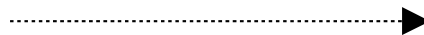


Gambar 3.12 Paket

- *Annotational Things*, merupakan bagian yang memperjelas model UML. Ia dapat berupa komentar-komentar yang menjelaskan fungsi serta ciri-ciri tiap elemen dalam model UML

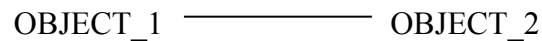
2. Relasi (*Relationship*), yang dimaksudkan adalah hubungan-hubungan yang terjadi antarelelemen dalam UML. Ada 4 macam relationship dalam UML sebagai berikut :

- *Dependency* (Kebergantungan), yaitu hubungan dimana perubahan yang terjadi pada suatu elemen independen akan mempengaruhi elemen yang bergantung padanya., simbol dependency :



Gambar 3.13 Simbol Dependency (Nugroho, 2005)

- Asosiasi, yaitu penghubung antara satu objek dengan objek yang lain. Suatu bentuk asosiasi adalah asosiasi yang menampilkan hubungan suatu objek dengan bagian-bagiannya, simbolnya adalah :



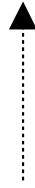
Gambar 3.14 Simbol Asosiasi (Nugroho, 2005)

- Generalisasi, yaitu hubungan dimana objek anak (descendent) berbagi perilaku dan struktur data dari objek yang ada di atasnya (objek induk – ancestor), simbolnya adalah :



Gambar 3.15 Simbol Generalisasi (Nugroho, 2005)

- Realisasi, yaitu operasi yang benar-benar dilakukan oleh suatu objek, simbolnya :



Gambar 3.16 Simbol Realisasi (Nugroho, 2005)

3. Diagram, dalam UML diagram digunakan untuk memodelkan suatu sistem. Terdapat 9 (sembilan) diagram dalam pemodelan UML, yang masing-masing diagram memiliki fungsi dan spesifikasi yang berbeda. Adapun kesembilan diagram tersebut adalah :
  - a) Diagram *Use Case*. Diagram ini memperlihatkan himpunan use case dan aktor-aktor. Fungsi utama diagram ini adalah untuk mengorganisasi dan memodelkan perilaku dari suatu sistem yang dibutuhkan dan diharapkan oleh pengguna.
  - b) Diagram kelas, yaitu diagram yang memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi. Akan tetapi walaupun bersifat statis diagram ini sering pula memuat kelas-kelas aktif.
  - c) Diagram objek, yaitu diagram yang memperlihatkan objek-objek dan relasi antar objek. Diagram objek memperlihatkan instansiasi statis dari segala sesuatu yang dijumpai pada diagram kelas.
  - d) Diagram urutan / *sequence diagram*, yaitu diagram interaksi yang menekankan pada pengiriman pesan (*message*) dalam satuan waktu tertentu.
  - e) Diagram kolaborasi, yaitu diagram interaksi yang menekankan organisasi struktural dari objek-objek yang menerima dan mengirim pesan (*message*)

- f) Diagram statechart, yaitu diagram yang memperlihatkan state-state pada sistem, memuat state, transisi, event, serta aktifitas. Fungsi utamanya adalah untuk memperlihatkan sifat dinamis dari antarmuka, kelas, kolaborasi dan pemodelan sistem-sistem yang reaktif.
- g) Diagram aktivitas, yaitu tipe khusus dari diagram state yang memperlihatkan aliran dari suatu aktifitas ke aktifitas ke aktifitas lainnya dalam suatu sistem. Fungsi utama dari diagram aktivitas adalah memodelkan fungsi-fungsi dalam suatu sistem dan memberikan tekanan pada aliran kendali antarobjek.
- h) Diagram komponen, diagram ini memperlihatkan organisasi serta kebergantungan pada komponen-komponen yang telah ada sebelumnya. Diagram ini berhubungan dengan diagram kelas dimana komponen secara tipikal dipetakan kedalam satu atau lebih kelas-kelas, antarmuka-antarmuka, serta kolaborasi-kolaborasi.
- i) Diagram penguraian/*deployment*, diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan (*run-time*). Diagram ini memuat simpul-simpul (*nodes*) beserta komponen-komponen yang ada didalamnya. Diagram ini memiliki kaitan erat dengan diagram komponen dimana deployment diagram memuat satu atau lebih komponen-komponen. Diagram ini sangat berguna saat aplikasi kita berlaku sebagai aplikasi yang dijalankan pada banyak mesin (*distributed computing*)



### 3.7 Arsitektur MVC

MVC (*Model-View-Controller*), merupakan sebuah teknik atau strategi dalam melakukan pengembangan aplikasi berbasis web. Ide dasar dari MVC ini adalah memisahkan pengendali logika dan pengendali tampilan (*separates business logic from presentation*). Teknik lain yang sejenis adalah MVP (*Model-Visual-Presentation*). Jika kita melihat MVC dari sudut pandang *separate business logic from presentation*. Maka *business logic*-nya ada di *model* dan *controller* sementara *presentation* ditangani oleh *view*. Tujuan dari pemisahan ini adalah untuk mempercepat dalam proses pengembangan perangkat lunak. Hal tersebut dimungkinkan karena pemisahan kode-kode yang bertugas untuk melakukan pengolahan data (PHP, ASP, SQL, dsb), dan kode-kode untuk presentasi data atau informasi (HTML, JavaScript). Pemisahan tersebut membuat penataan kode program lebih rapi sehingga memudahkan untuk pengembangan dan *maintenance* serta *error debugging*.

#### 3.7.1 Definisi Teknis MVC

Secara teknis definisi dari arsitektur MVC adalah sebagai berikut :

1. *Model* – mengandung fungsi-fungsi untuk melakukan pengolahan data dan *request* pengguna serta fungsi-fungsi untuk berkomunikasi dengan database.
2. *View* – berisi kode-kode atau tag yang bertugas untuk menampilkan informasi pada browser.
3. *Controller* – berisi kode-kode yang merupakan jembatan komunikasi antara pengguna dengan logika aplikasi. Di controller inilah inputan user di verifikasi untuk kemudian diteruskan ke model untuk dieksekusi. Begitu juga dengan hasil olahan dari model akan

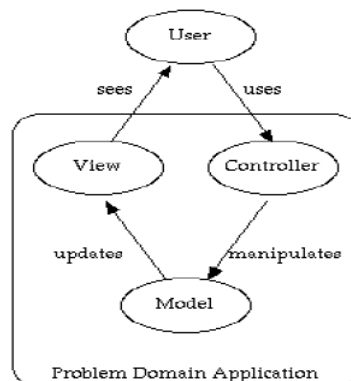
diterima terlebih dahulu oleh controler sebelum diteruskan ke view untuk ditampilkan di browser

### 3.7.2 Ilustrasi Kerja MVC

Dimisalkan pengguna mengklik sebuah link untuk melihat rekapitulasi keuangan harian.

Misalnya linknya seperti ini : *localhost/index.php?mod=rekap\_day&act=view*

1. HTTP request ini akan dikirim ke server dan diperiksa oleh sebuah file index.php dan dipilah-pilah.
2. Misalnya dengan modifikasi URL, diketahui bahwa *rekap\_day* adalah nama obyek CONTROLLER dan *view* adalah fungsi yg digunakan untuk menampilkan obyek tadi.
3. Maka dipanggilah file CONTROLLER *rekap\_day* dengan fungsi *view*
4. Pada CONTROLLER ini ada satu baris kode yang memanggil MODEL yang berisi query yang ada di file lain. Query ini bertugas menampilkan data yang diinginkan
5. Data hasil query oleh kode di CONTROLLER akan diperiksa dan dikirimkan ke file-file yang bertugas menampilkan data hasil query ini. File-file yg bertugas menampilkan data ini disebut VIEW.



Gambar 3.17 Pola Rancangan MVC (Cradinsky, 2001)

### 3.7.3 Template

Salah satu bagian penting dari konsep pengembangan aplikasi berbasis web dengan arsitektur model-view-controller adalah *templating*. *Template* membantu *developer* membagi *business logic* dari layout dan *content* dari aplikasi. Dengan menggunakan *template*, programmer tidak perlu menambahkan kode PHP atau script server yang lain langsung ke dalam kode HTML. Sebagai gantinya programmer hanya perlu menyisipkan *placeholder* dalam dokumen HTML, yang akan diganti dengan nilai sebenarnya oleh bahasa pemrograman yang berjalan di aplikasi server. *Template* merupakan bagian dari halaman HTML yang dapat ditampilkan ke *browser* hanya dengan menggunakan *template engine*. Untuk mengidentifikasi bagian tertentu dari halaman HTML yang berupa *template*, *template engine* menggunakan tag-tag XML untuk memberikan nama yang unik dan atribut yang beragam untuk *template*. Ketika memarsing sebuah *template*, *template engine* membagi halaman HTML kedalam beberapa bagian dan memperlakukannya sebagai *template* yang berbeda. Dengan menggunakan API (*Application Programming Interface*) yang disediakan oleh *template engine*, kita dapat menyembunyikan ataupun menampilkan bahkan *me-looping* *template-template* tertentu dalam halaman HTML. Dua dari sekian banyak *template engine* yang banyak digunakan dan tersedia gratis adalah *smarty* dan *patTemplate*.

## **BAB IV**

### **HASIL DAN PEMBAHASAN**

#### **4.1 Kebutuhan Sistem**

Agar dapat menjalankan dengan baik sistem manajemen keuangan yang dibangun. Maka sistem komputer yang digunakan harus memenuhi syarat minimal konfigurasi hardware dan software sebagai berikut :

##### **4.1.1 Kebutuhan Perangkat Keras**

Agar dapat berjalan dengan baik, komputer yang menjalankan sistem ini harus memenuhi system requirements minimal yang dibutuhkan. Spesifikasi dari system requirements tersebut adalah :

- 1.Processor Pentium III atau setara dengan kecepatan 667 MHz
- 2.Memori (RAM) dengan kapasitas 256 MB
- 3.Ruang kosong Harddisk sebesar 30 MB
- 4.Monitor VGA atau SVGA dengan resolusi minimum 1024 x 768.

##### **4.1.2 Kebutuhan Perangkat Lunak**

Ada beberapa perangkat lunak yang harus terinstal dalam sistem operasi komputer untuk dapat menjalankan sistem informasi ini, antara lain :

- 1.Web Server Apache 1.3, atau versi yang lebih baru.
- 2.PHP 4.2.x atau versi yang lebih baru
- 3.DBMS MySQL 4.2.x atau versi yang lebih baru yang mendukung innodb storage engine.

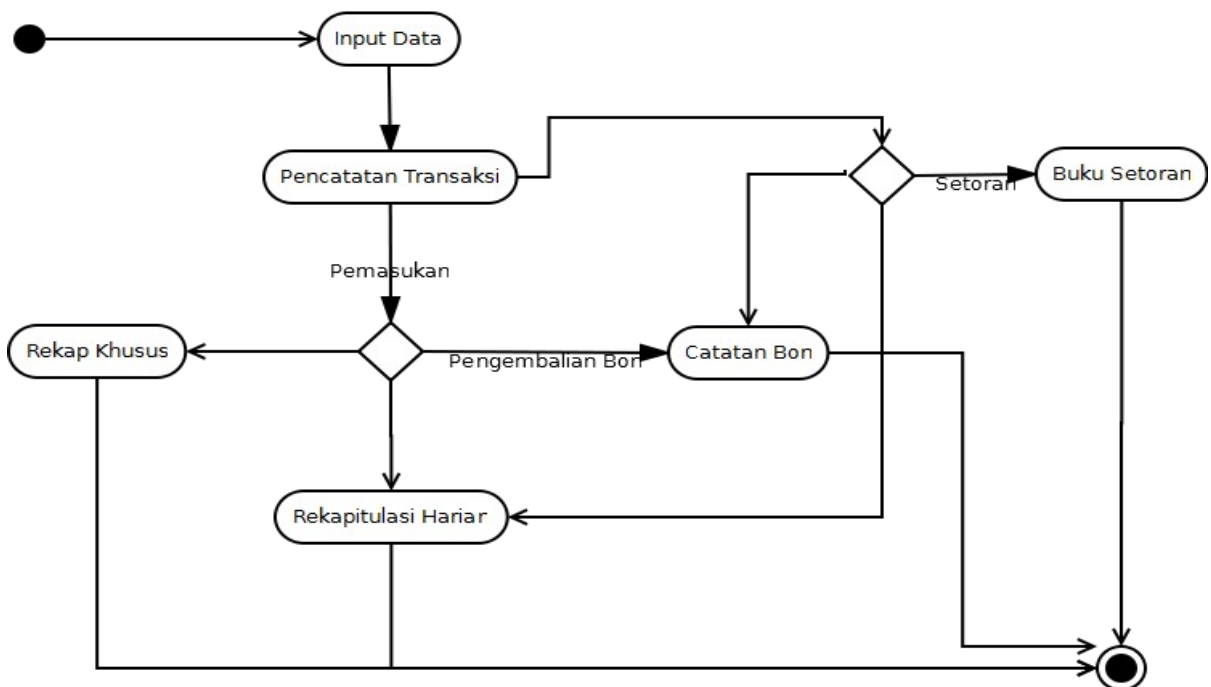
## 4.2 Metodologi Pengembangan Sistem

Dalam pengembangan sistem manajemen keuangan ini, digunakan metodologi yang mencakup analisa kebutuhan dan pemodelan sistem.

### 4.2.1 Analisa Kebutuhan / *Requirement Capturing*

Analisis kebutuhan dilakukan dengan mengidentifikasi bisnis proses yang terjadi dalam sistem yang akan dibangun. Pada tahap analisis, digunakan diagram aktifitas untuk menggambarkan proses apa saja yang terjadi di dalam sistem. diagram aktifitas merupakan salah satu diagram yang digunakan untuk menggambarkan alur kerja sistem. diagram aktifitas serupa dengan flowchart yang mendeskripsikan rangkaian perubahan dari satu aktivitas ke aktivitas yang lain.

Gambar 4.1 menunjukkan gambaran bisnis proses sistem yang dibangun menggunakan diagram aktifitas



Gambar 4.1 Diagram aktifitas proses bisnis sistem

Karena menggambarkan aliran kerja aktifitas, diagram aktifitas sangat cocok untuk menggambarkan bisnis proses dari suatu sistem. Selain itu, diagram aktifitas juga dapat menggambarkan aktivitas yang dilakukan oleh operator dalam hal ini manusia, maupun aktivitas yang dilakukan oleh mesin.

Dalam sistem informasi manajemen keuangan yang dibangun administrator dalam hal ini petugas keuangan merupakan aktor utama yang terlibat dalam sistem. Administratorlah yang akan melakukan pencatatan setiap transaksi dan memasukkannya kedalam pencatatan-pencatatan atau rekapitulasi-rekapitulasi yang lebih khusus.

Langkah pertama administrator akan melakukan pencatatan tiap-tiap transaksi yang terjadi di buku catatan sementara. Setelah itu dipilah mana yang merupakan dana masuk dan mana yang merupakan dana keluar. Bila transaksi itu berupa dana masuk maka dipilah kembali apakah dana itu berupa pengembalian bon/pinjaman atau pemasukan iuran maupun non iuran jika itu adalah dana pengembalian bon/pinjaman maka akan dimasukkan kedalam catatan bon/pinjaman. Jika dana itu berupa dana pemasukan iuran ataupun non iuran maka akan dicatat dalam buku rekap khusus untuk tiap iuran dan pemasukan non iuran.

Apabila transaksi itu berupa transaksi dana keluar dipilah kembali apakah dana keluar itu berupa setoran ke bendahara bidang atau berupa pinjaman/bon. Bila transaksi keluar tersebut adalah penyetoran dana ke bendahara bidang maka akan dicatat kedalam buku catatan setoran, namun bila transaksi itu merupakan pinjaman/bon maka akan dicatat dalam buku catatan pinjaman/bon.

Selanjutnya petugas/administrator akan melakukan rekapitulasi atau pencatatan dan penghitungan dana masuk dan dana keluar pada hari itu untuk mengetahui jumlah saldo yang masih tersedia.

Dari gambaran di atas, dapat diketahui masukan sistem, keluaran sistem, kebutuhan perangkat keras, kebutuhan perangkat lunak serta antar muka sistem yang akan dibuat, sehingga sistem yang dibuat nantinya sesuai dengan apa yang diharapkan.

#### **4.2.2 Pemodelan Sistem**

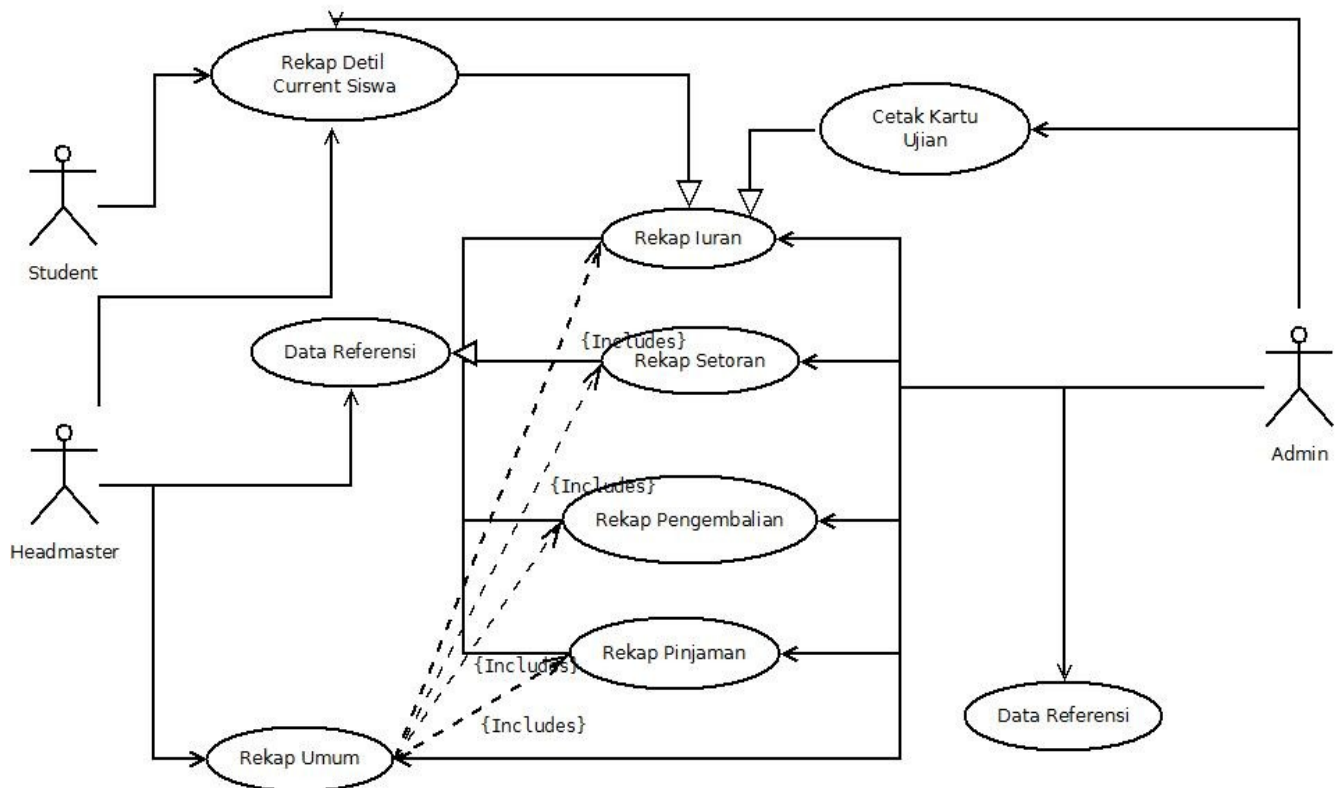
Analisis berorientasi objek (OOA) dilakukan untuk mendapatkan pemahaman yang lebih baik mengenai sistem dan persyaratan fungsionalnya. Dengan kata lain, OOA mengharuskan kita untuk mengidentifikasi kebutuhan fungsionalitas sistem dari perspektif pengguna dan mengidentifikasi objek, atribut data objek, behavior yang diasosiasikan, dan hubungan-hubungan yang mendukung fungsionalitas sistem yang diharapkan (Whitten CS, 2004). Untuk melakukan pemodelan sistem informasi manajemen keuangan ini digunakan tiga diagram Uml yaitu use case diagram, diagram urutan / sequence diagram, serta diagram kelas / class diagram.

##### **4.2.2.1. Diagram Uses Case**

Dari analisis yang telah dilakukan di atas, selanjutnya dibuat pemodelan dari sudut pandang user menggunakan *use case diagram*.

*Use case diagram* sistem manajemen keuangan yang dibangun ini menunjukkan peran masing-masing aktor dalam proses bisnis pencatatan iuran siswa di SMK Negeri 2 Temanggung. Dalam bisnis proses yang terjadi terdapat 3 (tiga aktor) yaitu administrator yang memiliki akses terhadap semua fitur yang disediakan oleh sistem. Kepala sekolah yang memiliki akses terhadap beberapa fitur kunci dari sistem yang meliputi rekapitulasi keuangan umum dan rekapitulasi iuran masing-masing siswa. Serta siswa yang hanya diperbolehkan mengakses rekapitulasi iurannya sendiri dan rekapitulasi pemasukan iuran secara umum.

Use case diagram untuk pemodelan sistem informasi manajemen keuangan ditunjukkan oleh gambar 4.2 sebagai berikut :



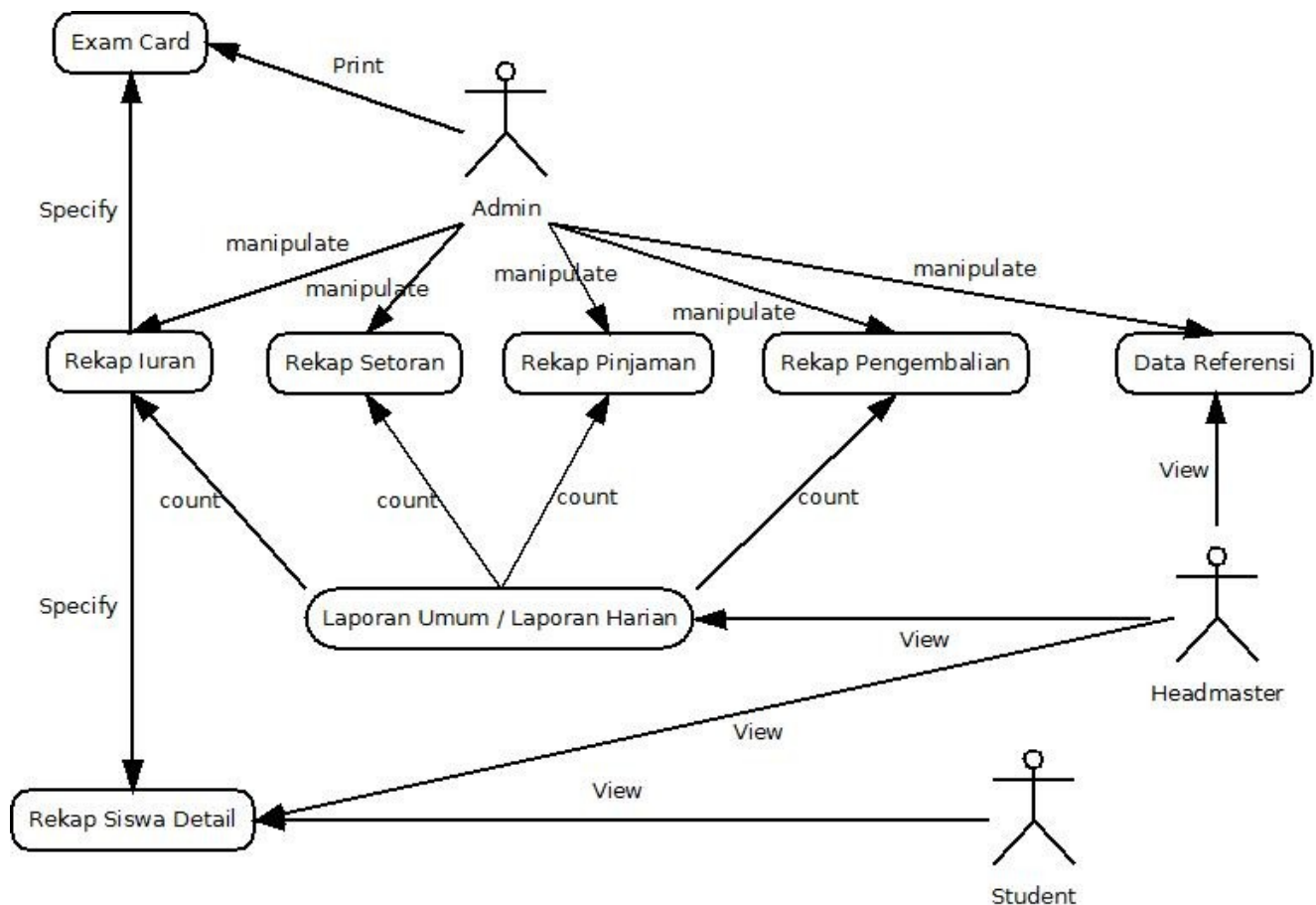
Gambar 4.2 Use case diagram sistem

Use case diagram di atas menggambarkan aktor-aktor dan aksi yang dilakukan oleh masing-masing aktor pada sistem. Use case masih bersifat umum dan belum menjelaskan transisi yang terjadi di dalam sistem. Oleh karena itu, use case diagram di atas perlu diuraikan lagi menjadi diagram kolaborasi / collaboration diagram.



#### 4.2.2.2. Diagram Kolaborasi / *Colaboratiao*n Diagram

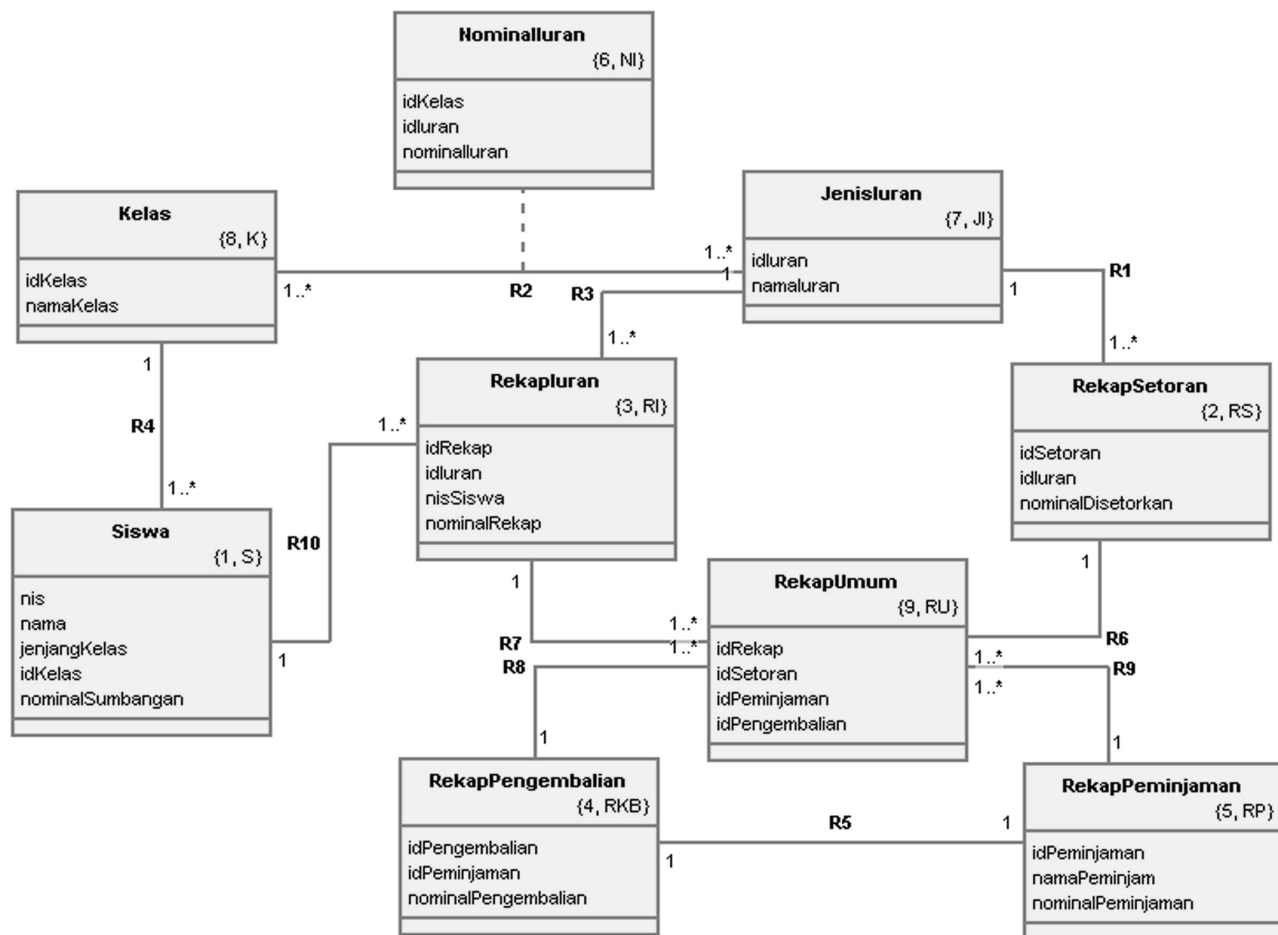
Dengan diagram kolaborasi kita dapat mengetahui urutan kondisi dimana antar elemen sistem saling berinteraksi. Hal ini dimungkinkan karena sequence diagram menekankan pada peran masing-masing objek (Nugroho, 2005). Diagram kolaborasi pada sistem informasi manajemen keuangan ini merupakan bentuk penjabaran detail dari use case diagram. Pada *use case diagram*, menggambarkan tentang aktor dan aktivitas aktor tersebut. Pada diagram kolaborasi aktivitas tersebut dijabarkan dengan menggambarkan peranan masing-masing objek terhadap objek yang lain. Diagram kolaborasi dari sistem yang dibangun dapat digambarkan sebagai berikut :



Gambar 4.3 Diagram Kolaborasi Sistem Manajemen Keuangan

#### 4.2.2.3. Diagram Kelas / Class Diagram

Dalam sistem yang dibangun ini ada beberapa class yang dibangun class-class ini memiliki keterkaitan satu sama lain. Adapun diagram kelas dari Sistem Manajemen Keuangan ini adalah sebagai berikut :



Gambar 4.4. Diagram Kelas sistem Manajemen Keuangan

Dari class diagram diatas dapat diketahui hubungan antar objek yang menjadi elemen-elemen penyusun dari sistem informasi yang dibangun. Dan dari hubungan antar objek tersebut maka didapatkan pula gambaran mengenai struktur utama dari basis data yang menjadi inti dari pengembangan sistem informasi manajemen keuangan ini.

#### 4.2.2.4. Desain Tabel Basis Data

Dari diagram kelas yang digambarkan pada point 4.2.3, maka komposisi dan struktur tabel yang menyusun basis data dari aplikasi sistem informasi keuangan yang dikembangkan adalah sebagai berikut.

a) Tabel siswa (sis\_siswa)

Tabel dibawah ini berfungsi untuk menyimpan data-data siswa meliputi Nomor Induk, nama, kelas, dan juga nominal sumbangan.

Tabel 4.1 Tabel Siswa

Nama Kolom	Tipe Data	Constraint
nisSiswa	integer	Primary key
namaSiswa	varchar(32)	
password	varchar(32)	
groupId	tinyint	
jenjangKelas	tinyint	
idKelas	tinyint	Foreign Key to Kelas
nominalSumbangan	longint	

b) Tabel kelas (sis\_kelas)

Tabel ini berfungsi menyimpan data jenis-jenis jurusan yang ada di SMK N 2 Temanggung

Tabel 4.2 Tabel Kelas

Nama Kolom	Tipe Data	Constraint
idKelas	smallint	Primary Key
namaKelas	varchar(32)	

c) Tabel nama iuran (sis\_nama\_iuran)

Merupakan tabel yang menyimpan data-data iuran meliputi nam iuran dan syarat berapa persenkah iuran harus dilunasi oleh siswa sebagai syarat ujian baik UTS maupun UAS.

Tabel 4.3 Tabel Nama Iuran

Nama Kolom	Tipe Data	Constraint
idNamaIuran	smallint	Primary Key
namaIuran	varchar(32)	
syaratUTS	smallint	
syaratUAS	smallint	
jenisIuran	varchar(32)	
isGeneral	enum	

d) Tabel nominal iuran (sis\_nominal\_iuran)

Tabel ini memiliki relasi dengan tael nama iuran dimana pada tabel ini informasi nominal iuran tiap jenis iuran untuk setiap jenjang kelas disimpan

Tabel 4.4 Tabel Nominal Iuran

Nama Kolom	Tipe Data	Constraint
idNominalIuran	smallint	Primary Key
idNamaIuran	smallint	Foreign Key to nama iuran
jenjangKelas	tinyint	
nominalIuran	longint	

e) Tabel peminjaman (sis\_peminjaman)

Tabel dibawah ini menyimpan data-data peminjaman dana yang terjadi meliputi nama peminjam, jumlah dana yang dipinjam dan waktu peminjaman

Tabel 4.5 Tabel Peminjaman

Nama Kolom	Tipe Data	Constraint
idPeminjaman	integer	Primary Key
namaPeminjam	varchar(32)	
nominalPeminjaman	longint	
nominalTelahDibayar	longint	
tanggalPeminjaman	date	

f) Tabel pengembalian (sis\_pengembalian)

Tabel ini memiliki relasi dengan tabel peminjaman, berisikan data-data pengembalian dana untuk setiap peminjaman yang tercatat pada tabel peminjaman.

Tabel 4.6 Tabel Pengembalian

Nama Kolom	Tipe Data	Constraint
idPengembalian	integer	Primary Key
idPeminjaman	integer	Foreign Key to idPeminjaman
nominalPengembalian	longint	
tanggalPengembalian	date	

g) Tabel rekap iuran (sis\_rekap\_iuran)

Tabel ini berfungsi untuk menyimpan setiap transaksi pembayaran iuran yang terjadi. Data yang dicatat meliputi identitas siswa yang membayar, jenis dan nominal iuran yang dibayarkan, serta tanggal pembayaran.

Tabel 4.7 Tabel Rekap Iuran Siswa

Nama Kolom	Tipe Data	Constraint
idRekapIuran	integer	Primary Key
nisSiswa	integer	Foreign Key to Siswa
idNamaIuran	tinyint	Foreign Key to Nama Iuran
nominalRekapIuran	longint	
tanggalRekap	date	

h) Tabel rekap setoran (sis\_rekap\_setoran)

Tabel ini berfungsi untuk menyimpan setiap transaksi penyetoran iuran yang terjadi. Data yang dicatat meliputi jenis dan nominal iuran yang disetorkan, serta tanggal penyetoran.

Tabel 4.8 Tabel Rekap Setoran

Nama Kolom	Tipe Data	Constraint
idRekapSetoran	integer	Primary Key
idNamaIuran	tinyint	Foreign Key to Nama Iuran
nominalRekapSetoran	longint	
tanggalPenyetoran	date	
keterangan	text	

### 4.3 Implementasi Sistem

Dalam implemetasi di komputer sistem informasi ini tersusun dari beberapa modul yang digunakan untuk proses pengolahan informasi ataupun hal-hal yang mendukung proses tersebut dan tidak berkaitan langsung dengan pengolahan informasi. Tiap satu modul memiliki satu shortcut menu dalam user interface yang berfungsi untuk menjembatani interaksi user dengan aplikasi tersebut.

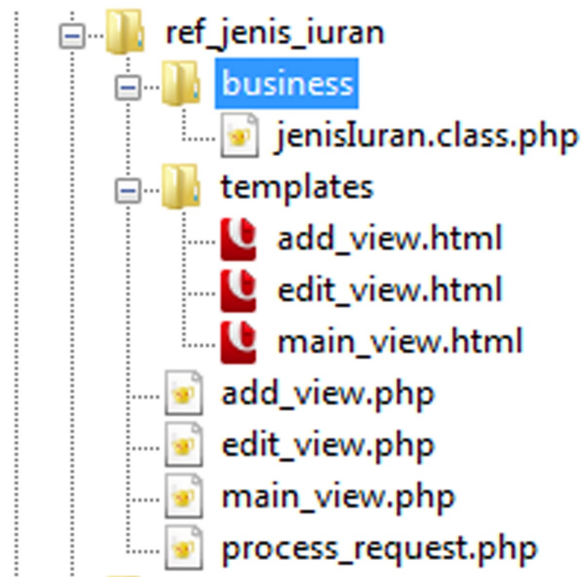
Berikut adalah daftar modul yang menyusun sistem informasi manajemen keuangan ini :

1. ref\_jenis\_iuran : modul untuk manage referensi jenis iuran
2. ref\_data\_siswa : modul untuk mengolah data-data siswa
3. iuran\_bulanan\_siswa : modul untuk melakukan input pembayaran iuran bulanan
4. iuran\_tahunan\_siswa : modul untuk melakukan input pembayaran iuran tahunan
5. iuran\_dps : modul untuk melakukan pengelolaan cicilan sumbangan.
6. rekap\_setoran : modul yang memiliki fungsi manage data-data setoran iuran
7. rekap\_pinjaman : modul yang berfungsi memanipulasi data peminjaman dan pengembalian.
8. rekap\_siswa : modul yang berfungsi untuk memanipulasi rekap iuran siswa

9. rekap\_umum : modul untuk menampilkan rekapitulasi umum (pengeluaran dan pemasukan)
10. black\_list : modul yang menyajikan informasi siswa yang tidak bisa mengikuti UTS/UAS
11. cek\_ijin\_ujian : modul untuk mengecek perijinan ujian siswa.
12. ganti password : modul untuk melakukan pergantian password

#### 4.3.1 Struktur File

Struktur file dari tiap-tiap modul yang menyusun aplikasi terdiri dari beberapa file php dan html, struktur file dari masing-masing modul adalah sama. Dapat digambarkan sebagai berikut :



Gambar 4.5 struktur file module aplikasi

Struktur file terdiri dari beberapa file dan dua sub modul yang juga didalamnya ada beberapa file. File yang terletak pada module bisnis merupakan class utama dari setiap module, berisikan bussiness logic dari aplikasi. Sedangkan file yang terletak pada folder templates merupakan file HTML yang akan menampilkan informasi ke browser. Setiap file HTML dalam folder tersebut



memiliki satu file controler yang letaknya langsung dibawah directori module. Misal file add\_view.html memiliki file controler yang bernama view.php. File controler yang bernama process\_request.php tidak memiliki file templates/view. File tersebut hanya berguna jika ada interaksi di file templates yang memparsing / mengirimkan input nilai ke class utama.

#### 4.3.2 *Library*

Dalam membangun sistem informasi manajemen keuangan ini digunakan library yang dibuat oleh pihak ketiga yang berfungsi untuk memberikan dukungan teknis terhadap proses bisnis yang berjalan. Aplikasi ini menggunakan empat *library* yaitu :

- **jQuery** : jQuery merupakan *library* JavaScript yang memiliki banyak fungsi JavaScript yang siap dipakai untuk pengembangan aplikasi berbasis web. Pada aplikasi ini jQuery digunakan untuk mengimplementasikan XMLHttpRequest atau yang lebih dikenal dengan Ajax pada saat penginputan nilai dari view ke class utama melalui form.
- **AES** : AES sebenarnya merupakan algoritma enkripsi. Pada aplikasi ini *library* AES digunakan untuk melakukan enkripsi parameter url yang berupa variable \$\_GET. Hal ini bertujuan untuk menghindari penyusupan atau akses ilegal kedalam basis data melalui manipulasi url seperti SQL Injection maupun PHP Injection.
- **patTemplate** : patTemplate merupakan library yang berfungsi untuk melakukan *templating* aplikasi web. *Library* ini sangat membantu dalam memisahkan business logic dari presentation.
- **WriteExcel** : Merupakan *library* yang awalnya ditulis dengan bahasa Perl-CGI untuk merepresentasikan nilai-nilai dari variabel dalam bahasa pemrograman kedalam bentuk *spreadsheet*. Dalam perkembangannya *library* ini juga didukung oleh bahasa pemrograman PHP mulai PHP versi 4.1.1

### 4.3.3 Model-View-Controller Sistem

Komponen dan modul dibuat dengan menggunakan *design pattern* MVC (*Model-View-Controller*). Untuk satu modul paling tidak terdiri dari satu *Model*, satu *Controller* dan satu *View*.

File Model berupa *class* yang berhubungan dengan pengambilan data dari database. *Class* tersebut memiliki beberapa *method* yang digunakan untuk melakukan proses manipulasi tabel database. Berikut contoh kelas untuk modul ref\_jenis\_iuran.

```
class jenisIuran extends mysql_db{

    function view_all(){
        return $this->execute("SELECT idJenisIuran AS Id,
                                namaJenisIuran AS Nama,
                                JenisIuran AS Jenis,
                                IF(NominalJenjangKelas, NominalJenjangKelas, '1-2-3') AS Jenjang,
                                nominalIuran AS Nominal, CONCAT(syaratUTS,'%') AS
                                Syarat_UTS, CONCAT(syaratUAS,'%') AS Syarat_UAS
                                FROM
                                sis_nama_iuran
                                JOIN sis_nominal_iuran ON idNominalJenisIuran=idJenisIuran");
    }

    function get_iuran($idEd, $jenjang){
        if($jenjang=='1-2-3') $jenjang=0;
        return $this->execute("SELECT idJenisIuran AS Id,
                                namaJenisIuran AS Nama,
                                JenisIuran AS Jenis,
                                NominalJenjangKelas AS Jenjang,
                                nominalIuran AS Nominal,
                                syaratUTS AS Syarat_UTS,
                                syaratUAS AS Syarat_UAS
                                FROM
                                sis_nama_iuran
                                JOIN sis_nominal_iuran ON idNominalJenisIuran=idJenisIuran
                                WHERE idJenisIuran='$idEd' AND NominalJenjangKelas='$jenjang'");
    }
}
```

*Class* diatas digunakan untuk memanipulasi data dari basis data dalam modul ref\_jenis\_iuran. Semua *Class* utama pada tiap-tiap modul merupakan anak dari class mysql\_db dimana dalam class tersebut didefinisikan konfigurasi untuk melakukan koneksi dan representasi data dalam bentuk array .

Method-method yang terdapat dalam class tersebut merupakan kumpulan business logic yang digunakan untuk memanipulasi data dan menyajikannya dalam bentuk informasi.

Untuk menjembatani interaksi user dengan logika aplikasi pada class utama tersebut maka dibuatlah file controller. File *controller* menginisialisasi model dengan membuat objek dari class *jenisIuran*. Pada arsitektur MVC, untuk *memparsing* data model ke view harus dilakukan menggunakan *template engine* (*patTemplate*). Contoh file controller yang terdapat dalam modul *ref\_jenis\_iuran* adalah sebagai berikut :

```
<?php
require_once("business/jenisIuran.class.php");

$tpl->setBasedir("module/ref_jenis_iuran/templates");
$tpl->readTemplatesFromFile("main_view.html");

$urlAdd=$crypt->paramEncrypt('module=ref_jenis_iuran&file=add_view');
$tpl->addVar("article", "HEADLINE", "Jenis Iuran Siswa");
$tpl->addVar("article", "URL_TAMBAH", "index.php?". $urlAdd);

$view=new jenisIuran;
$view->connect();
$view->view_all();

$row=0;
while($result=$view->get_array()){
    $number=$row+1;
    if ($number % 2 ==0){
        $class_name='table-common-even';
    }else{
        $class_name='';
    }
    $disp[$row]=array(
        'NO'=>$number, 'ID'=>$result['Id'], 'NAMA'=>$result['Nama'], 'JENJANG'=>$result['Jenjang'],
        'JENIS'=>$result['Jenis'], 'NOMINAL'=>number_format($result['Nominal'],2,',','.'),
        'UTS'=>$result['Syarat_UTS'], 'UAS'=>$result['Syarat_UAS'], 'CLASS_NAME'=>$class_name);
    $row++;
}
#print_r($disp);
foreach ($disp as $data){
    $param="module=ref_jenis_iuran&file=edit_view&idEd=".$data[ID]. "&jenjang=".$data[JENJANG];
    $urlEdt=$crypt->paramEncrypt($param);

    $tpl->addVars("data", $data);
    $tpl->addVar("data", "URL_EDIT", "index.php?". $urlEdt);
    $tpl->parseTemplate("data", "a");
}
?>
```

File *View* ditentukan pada inisiasi method `$tmpl->readTemplateFromBaseDir` . View yang dipakai adalah file html. Berikut source code untuk view `ref_jenis_iuran`.

```

<!-- patTemplate:tmpl name="article" -->
<h3>{HEADLINE}</h3>
<table class="table-common">
  <tr>
    <th>No.</th><th>Nama Iuran</th><th>Jenis</th><th>Kelas</th>
    <th>Nominal</th><th>Syarat UTS</th><th>Syarat UAS</th><th colspan=2>Aksi</th>
  </tr>
  <!-- patTemplate:tmpl name="data" -->
  <tr class={CLASS_NAME}>
    <td>{NO}</td><td>{NAMA}</td><td>{JENIS}</td><td>{JENJANG}</td>
    <td>{NOMINAL}</td><td>{UTS}</td><td>{UAS}</td>
    <td><a href='{URL_EDIT}' title="Ubah"></a></td>
    <td><a href='{URL_HAPUS}' title="Hapus"></a></td>
  </tr>
<!-- /patTemplate:tmpl -->
</table>
<div class="toolbar">
  <a href='{URL_TAMBAH}' title="Tambah Data"></a>
</div>
<!-- /patTemplate:tmpl -->

```

Tampilan yang dimunculkan pada browser adalah *skrip* yang ada diantara `<patTempalte:tmpl name="article">` dan `</patTemplate:tmpl>`. Sedangkan *string* yang ada diantara kurung kurawal, akan diisi dengan data hasil *parsing* pada *controller*.

#### 4.3.4 Implementasi User Interface

Pada bagian ini akan ditampilkan beberapa halaman yang merupakan fitur-fitur utama dari sistem informasi keuangan yang dibangun.

##### 1. Halaman Referensi

Halaman merupakan salah satu halaman dalam menu manajemen referensi yang berfungsi untuk melakukan pengolahan jenis iuran yang dijadikan rujukan untuk pengolahan data transaksi.

NO.	NAMA IURAN	JENIS	KELAS	NOMINAL	SYARAT UTS	SYARAT UAS	AKSI
1	Dana Komite	Bulanan	1	125.000,00	50%	100%	
2	Dana Komite	Bulanan	2	100.000,00	50%	100%	
3	Dana Komite	Bulanan	3	75.000,00	50%	100%	
4	Dana Tabungan Akhir	Bulanan	1-2-3	25.000,00	0%	0%	
5	Dana Prakerin	Tahunan	1-2-3	75.000,00	0%	100%	
6	Dana Kesiswaan	Tahunan	1-2-3	75.000,00	0%	100%	
7	Dana Pengembangan Sekolah	Sumbangan	1-2-3	0,00	0%	100%	

+ Tambah

copyright © muhammad rifqi ma'arif  
informatics engineering state islamic university of jogjakarta  
muhammad.rifqi@gmail.com

Gambar 4.7 Halaman Referensi Jenis Iuran Siswa

## 2. Halaman Manajemen Pemasukan

Halaman pada menu manajemen pemasukan berfungsi untuk melakukan input transaksi pembayaran yang terjadi. Terdiri dari tiga halaman utama yaitu halaman pemasukan iuran bulanan, halaman pemasukan iuran tahunan dan halaman pemasukan dana sumbangan. Gambar dibawah ini merupakan halaman pemasukan iuran bulanan, dua halaman yang lain memiliki format yang kurang lebih serupa

The screenshot displays the 'Sistem Informasi Keuangan' (Financial Information System) interface for SMK Negeri 2 Temanggung. The header includes the school logo and name, along with 'Depan' and 'Logout' links. The main title is 'Sistem Informasi Keuangan' with the subtitle 'PENGELOLAAN DANA IURAN SISWA'. A left sidebar menu lists various management options under categories like 'Manajemen Referensi', 'Manajemen Pemasukan', 'Manajemen Dana', 'Rekapitulasi Data', 'Perijinan Ujian', and 'Authentikasi'. The main content area is titled 'Pemasukan Iuran Bulanan Siswa' and contains a form with the following fields: 'NIS' (text input), 'Nominal' (text input), and 'Jenis Pembayaran' (checkboxes for 'Dana Komite' and 'Dana Tabungan Akhir', each followed by a month selection dropdown and the word 'Bulan'). At the bottom of the form are 'Submit' and 'Reset' buttons. The footer contains copyright information for Muhammad Rifqi Ma'arif.

SMK Negeri 2 Temanggung

Depan Logout

**Sistem Informasi Keuangan**  
PENGELOLAAN DANA IURAN SISWA

**:: Manajemen Referensi**  
Referensi Jenis Iuran  
Referensi Data Siswa  
Setting Semester

**:: Manajemen Pemasukan**  
Iuran Bulanan Siswa  
Iuran Tahunan Siswa  
Dana Pengembangan

**:: Manajemen Dana**  
Manajemen Setoran  
Peminjaman Dana

**:: Rekapitulasi Data**  
Rekapitulasi @siswa  
Rekapitulasi Umum

**:: Perijinan Ujian**  
Lihat Black List  
Cetak Kartu Ujian

**:: Autentikasi**  
Ganti Password

**Pemasukan Iuran Bulanan Siswa**

NIS :

Nominal :

Jenis Pembayaran : ☐ Dana Komite -  Bulan  
☐ Dana Tabungan Akhir -  Bulan

copyright @ muhammad rifqi ma'arif  
informatics engineering state islamic university of jogjakarta  
muhammad.rifqi@gmail.com

Gambar 4.8 Halaman Pemasukan Iuran Bulanan

### 3. Halaman Manajemen Dana

Halaman manajemen dana merupakan halaman yang fungsinya untuk melakukan manajemen pengeluaran dana yaitu penyetoran dana dan peminjaman dana fungsi lainnya dari halaman ini adalah melakukan pencatatan pengembalian atau cicilan pinjaman. Kedua halaman tersebut ditunjukkan oleh gambar 4.9 dan 4.10 dibawah ini :

The screenshot displays the 'Manajemen Penyetoran Dana' (Fund Deposit Management) interface. The header includes the school logo and name 'SMK Negeri 2 Temanggung', along with 'Depan' and 'Logout' links. The main title is 'Sistem Informasi Keuangan' with the subtitle 'PENGELOLAAN DANA IURAN SISWA'. The sidebar menu on the left lists various system functions. The main form area is titled 'Manajemen Penyetoran Dana' and contains the following fields:

- Nominal Setoran :** A text input field for the deposit amount.
- Jenis Pembayaran :** A section with five radio button options:
  - ☐ Setoran Dana Komite
  - ☐ Setoran Dana Tabungan Akhir
  - ☐ Setoran Dana Prakerin
  - ☐ Setoran Dana Kesiswaan
  - ☐ Setoran Dana Pengembangan Sekolah
- Keterangan :** A text input field for additional notes.

At the bottom of the form are two buttons: 'Submit' and 'Reset'.

Gambar 4.9 Halaman Penyetoran Dana

 SMK Negeri 2 Temanggung

Depan Logout

# Sistem Informasi Keuangan

PENGLOLAAN DANA IURAN SISWA

:: Manajemen Referensi

Referensi Jenis Iuran

Referensi Data Siswa

Setting Semester

:: Manajemen Pemasukan

Iuran Bulanan Siswa

Iuran Tahunan Siswa

Dana Pengembangan

:: Manajemen Dana

Manajemen Setoran

Peminjaman Dana

:: Rekapitulasi Data

Rekapitulasi @siswa

Rekapitulasi Umum

:: Perijinan Ujian

Lihat Black List

Cetak Kartu Ujian

:: Autentikasi

Ganti Password

## Tambah Data Peminjaman

Nama Peminjam :

Jumlah Dipinjam :

copyright © muhammad rifqi ma'arif  
informatics engineering state islamic university of jogjakarta  
muhammad.rifqi@gmail.com


#### 4.10 Halaman Input Peminjaman

Selain kedua halaman tersebut pada menu manajemen dana ini ada beberapa tambahan halaman yaitu halaman untuk melakukan rekapitulasi penyetoran dana, penyuntingan histori penyetoran dan halaman untuk melakukan manajemen dana peminjaman.



#### 4. Halaman Rekapitulasi Data

Halaman rekapitulasi data ini merupakan halaman yang memuat tentang rekapitulasi data pembayaran yang telah dilakukan oleh masing-masing siswa dari tiap-tiap kelas. Selain itu menu ini juga memiliki kemampuan untuk menampilkan rekapitulasi data secara umum yang mencakup data pemasukan dan pengeluaran. Informasi rekapitulasi data keuangan siswa ditunjukkan oleh gambar 4.11 dibawah ini :

**SMK Negeri 2 Temanggung**

Depan Logout

**Sistem Informasi Keuangan**  
PENGELOLAAN DANA IURAN SISWA

**:: Manajemen Referensi**  
Referensi Jenis Iuran  
Referensi Data Siswa  
Setting Semester

**:: Manajemen Pemasukan**  
Iuran Bulanan Siswa  
Iuran Tahunan Siswa  
Dana Pengembangan

**:: Manajemen Dana**  
Manajemen Setoran  
Peminjaman Dana

**:: Rekapitulasi Data**  
Rekapitulasi @siswa  
Rekapitulasi Umum

**:: Perijinan Ujian**  
Lihat Black List  
Cetak Kartu Ujian

**:: Autentikasi**  
Ganti Password

Lihat Black List Ujian Akhir Semester

NO.	NIS	NAMA SISWA	TOTAL DANA IURAN YANG SUDAH DIBAYARKAN				
			KOMITE	TA	PRAKERIN	KESISWAAN	DPS
1	06650001	Abdullah	125.000,00	25.000,00	75.000,00	75.000,00	100.000,00
2	06650002	Ali	250.000,00	50.000,00	75.000,00	75.000,00	200.000,00
3	06650003	Abu Bakar	0,00	0,00	0,00	0,00	0,00
4	06650004	Usman	0,00	0,00	0,00	0,00	0,00
5	06650005	Umar	0,00	0,00	0,00	0,00	0,00
6	06650007	Saad	0,00	0,00	0,00	0,00	0,00
7	06650008	Abu Ubaidillah	0,00	0,00	0,00	0,00	0,00
8	06650009	Bukhari	0,00	0,00	0,00	0,00	0,00
9	06650010	Abdurrahman	0,00	0,00	0,00	0,00	0,00
10	06650011	Fatimah	0,00	0,00	0,00	0,00	0,00
11	06650012	Khatidjah	0,00	0,00	0,00	0,00	0,00
12	06650013	Marhamah	0,00	0,00	0,00	0,00	0,00
13	06650014	Maemunah	0,00	0,00	0,00	0,00	0,00
14	06650015	Mahmudah	0,00	0,00	0,00	0,00	0,00

Export Excel

copyright © muhammad rifqi ma'arif  
informatics engineering state islamic university of jogjakarta  
muhammad.rifqi@gmail.com

Gambar 4.11 Halaman Rekapitulasi Data Iuran siswa

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

1. Pelaksanaan kerja praktek ini berhasil menganalisis proses bisnis sistem keuangan khususnya pencatatan dan rekapitulasi iuran siswa di SMK N 2 Temanggung dan memodelkannya dalam bentuk berorientasi objek.
2. Hasil analisis dan pemodelan tersebut berhasil diimplementasikan dalam sebuah aplikasi berbasis web dengan skema atau arsitektur MVC (*model-view-controler*)
3. Sistem yang dibangun memiliki kemampuan untuk membantu petugas keuangan mengelola dana iuran siswa dengan cepat dan akurat.

#### **5.2 Saran**

1. Mengingat jenis iuran untuk tiap jenjang kelas berbeda dan banyaknya siswa, maka untuk pengembangan selanjutnya akan lebih baik jika ditambahkan fitur untuk melakukan update secara otomatis pada jenjang kelas siswa saat kenaikan kelas sehingga administrator tidak perlu melakukan update secara manual.
2. Sistem ini akan lebih lengkap jika ditambahkan fungsionalitas untuk pengelolaan dana di masing-masing bendahara iuran.

## DAFTAR PUSTAKA

- Booch, Graddy. Dkk. 2004. **The UML User Guide**. New York : Addison-Wilson
- Cradisnky, Leonard. 2001. **The Tier Architecture**. [www.ssuet.edu](http://www.ssuet.edu), tanggal akses 23 September 2009
- Diamond, Jack. Pokar, Khemani. 2005. **Introducing Financial Management System**.  
Geneva : International Monetary Fund
- Dorotinsky, Bill. 2003. **Implementing Financial Management Information System Projects : The World Bank Experience**. New York : The World Bank
- Gerkee, Johanness. Ramakhrisnan, Raguu. 2008. **Database Management System**.  
New York : McGraw Hill
- Jessup, Leonard. Valacich, Joseph. 2003. **Information System Today**. Prentice Hall
- Jogiyanto, 2001, “**Analisis dan Desain Sistem Informasi : Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis**”. Yogyakarta : Penerbit Andi
- Nugroho, Adi. 2005. **Perancangan Sistem Informasi dengan Metode Berorientasi Objek**.  
Bandung : Informatika
- Laudon, Laudon. 2006. **Management Information System**. New York : McGraw Hill
- Pender, Tom. 2003. **The UML Bible**. New York : John Wiley & Sons Publishing
- Powell, Gavin. 2005. **Beginning Database Design**. New York : John Wiley & Sons Publishing
- Robin-Brown, Edin. 2008. **Integrated Financial Management System**. Washington : USAID Doc
- Whitten, James. Dkk. 2004. **Metodologi Analisis dan Perancangan Sistem**.  
Yogyakarta : Andi Publisher

## Lampiran A

### Source Code File Model Modul Pemasukan Jenis Iuran

```
<?php
require_once($_SERVER['DOCUMENT_ROOT']."/siskeu_smk2/class/mysql_db.class.php")
class inputIuran extends mysql_db{
    function display_jenis_iuran(){
        return $this->execute("SELECT idJenisIuran AS ID, namaJenisIuran AS
                                NAMA FROM sis_nama_iuran WHERE JenisIuran =
                                'Bulanan'");
    }
    function get_nominal_iuran($kelas, $idIuran){
        $check=$this->execute("SELECT IsGeneral FROM sis_nama_iuran WHERE
                                idJenisIuran='$idIuran'");
        $checkResult=$this->get_array();
        if($checkResult=0){
            return $this->execute("SELECT nominalIuran AS Nominal FROM
                                    sis_nominal_iuran WHERE
                                    NominalJenjangKelas='$kelas' AND
                                    idNominalJenisIuran='$idIuran'");
        }else{
            return $this->execute("SELECT nominalIuran AS Nominal FROM
                                    sis_nominal_iuran WHERE
                                    idNominalJenisIuran='$idIuran'");
        }
    }
    function update_rekap_iuran($nis, $idIuran, $jml_bayar){
        return $this->execute("INSERT INTO sis_rekap_iuran VALUES
                                ('$nis', '$idIuran', '$jml_bayar', NOW(), '')");
    }
}
?>
```

## Lampiran B

### Source Code File Model Modul Pengeluaran Setoran

<?

```
require_once($_SERVER['DOCUMENT_ROOT']."/siskeu_smk2/class/mysql_db.class.php");

class Setoran extends mysql_db{

    function display_jenis_setoran(){

        return $this->execute("SELECT idJenisIuran AS ID, CONCAT('Setoran
        ',namaJenisIuran) AS NAMA FROM sis_nama_iuran");

    }

    function get_rekap_setoran(){

        return $this->execute("SELECT id AS ID, nama_setoran AS NAMA_SETORAN,
        SUM(jumlah_masuk) AS JUMLAH_MASUK,SUM(jumlah_disetorkan) AS
        JUMLAH_DISETORKAN, SUM(jumlah_masuk)-SUM(jumlah_disetorkan)
        AS JUMLAH_SISA FROM (SELECT idJenisIuran AS id,
        namaJenisIuran AS nama_setoran, IF(SUM(nominalRekap),
        SUM(nominalRekap),0) AS jumlah_masuk,0 AS jumlah_disetorka
        FROM sis_nama_iuran LEFT JOIN sis_rekap_iuran ON
        idJenisIuran=idIuranRekap GROUP BY idJenisIuran UNION
        SELECT idJenisIuran AS id, namaJenisIuran AS nama_setoran,
        0 AS jumlah_masuk, IF(SUM(nominalRekapSetoran),
        SUM(nominalRekapSetoran),0) AS jumlah_disetorkan FROM
        sis_nama_iuran LEFT JOIN sis_rekap_setoran ON
        idJenisIuran=idNamaRekapSetoran GROUP BY idJenisIuran ) tmp
        GROUP BY id ORDER BY id ASC");

    }

    function get_nama_setoran($idSetoran){

        return $this->execute("SELECT nama JenisIuran AS NAMA FROM
        sis_nama_iuran WHERE idJenisIuran='$idSetoran'");

    }

    function get_history_setoran_detil($idSetoran){

        return $this->execute("SELECT idRekapSetoran AS ID, tanggalPenyetoran
        AS TANGGAL nominalRekapSetoran AS TOTAL,keterangan AS
        KETERANGAN FROM sis_rekap_setoran WHERE idNamaRekapSetoran
        = $idSetoran");

    }

    function get_max_setoran($idSetoran){

        return $this->execute("SELECT SUM(rekap) - SUM(setoran) AS maks FROM
        ( SELECT SUM(nominalRekap) AS rekap, 0 AS setoran FROM
```

```

        sis_rekap_iuran WHERE idIuranRekap='$idSetoran' UNION
        SELECT 0 AS rekap, SUM(nominalRekapSetoran) AS setoran FROM
        sis_rekap_setoran WHERE idNamaRekapSetoran='$idSetoran')
        tmp");
    }

function get_detil_transaksi_st($idTr){
    return $this->execute("SELECT idRekapSetoran As ID_TR,
        tanggalPenyetoran AS TANGGAL, idNamaRekapSetoran AS ID_NAMA,
        nominalRekapSetoran AS NOMINAL, keterangan AS KETERANGAN FROM
        sis_nama_iuran JOIN sis_rekap_setoran ON idJenisIuran =
        idNamaRekapSetoran WHERE idRekapSetoran = '$idTr'");
}

function insert_setoran($idSetoran, $nominal, $ket){
    return $this->execute("INSERT INTO sis_rekap_setoran VALUES('`,`
        '$idSetoran', '$nominal', NOW(), '$ket')");
}

function update_setoran($idSt, $jmlTr, $tglTr, $ket, $idTr){
    return $this->execute("UPDATE sis_rekap_setoran SET idNamaRekapSetoran
=        '$idSt', nominalRekapSetoran = '$jmlTr', tanggalPenyetoran
        = '$tglTr', keterangan = '$ket' WHERE
        idRekapSetoran='$idTr'");
}

}

?>

```

## Lampiran C

### Source Code File Model Modul Rekapitulasi Iuran Siswa

<?

```
require_once($_SERVER['DOCUMENT_ROOT']."/siskeu_smk2/class/mysql_db.class.php");

class rekap_siswa extends mysql_db{

    function get_kelas_detil($klsId){

        return $this->execute("SELECT namaKelas AS NAMA FROM sis_kelas WHERE
                                idKelas='$klsId'");

    }

    function get_list_siswa($klsJenjang, $klsId){

        return $this->execute("SELECT nisSiswa AS NIS,namasiswa AS NAMA,(SELECT
                                SUM(nominalRekap) FROM sis_rekap_iuran WHERE idIuranRekap =
                                1 AND nisSiswaRekap=nisSiswa) AS TOTAL_KOMITE,(SELECT
                                SUM(nominalRekap)FROM sis_rekap_iuran WHERE idIuranRekap = 2 AND
                                nisSiswaRekap=nisSiswa) AS TOTAL_TA,(SELECT SUM(nominalRekap)FROM
                                sis_rekap_iuran WHERE idIuranRekap = 3 AND
                                nisSiswaRekap=nisSiswa) AS TOTAL_PRAKERIN,(SELECT
                                SUM(nominalRekap)FROM sis_rekap_iuran WHERE idIuranRekap = 4 AND
                                nisSiswaRekap=nisSiswa) AS TOTAL_OSIS,(SELECT
                                SUM(nominalRekap) FROM sis_rekap_iuran WHERE idIuranRekap
                                = 5 AND nisSiswaRekap=nisSiswa) AS TOTAL_DPS, nominalDPS AS
                                NoMINAL_DPS FROM sis_siswa LEFT OUTER JOIN sis_rekap_iuran
                                ON (nisSiswa=nisSiswaRekap) LEFT OUTER JOIN sis_nominal_dps
                                ON (nisSiswa=nisSiswaDps) WHERE jenjangKelasSiswa =
                                '$klsJenjang' AND idKelasSiswa = '$klsId' GROUP BY
                                nisSiswa");

    }

    function get_siswa($nis){

        return $this->execute("SELECT nisSiswa AS NIS, namaSiswa AS NAMA,
                                jenjangKelasSiswa AS JENJANG,namaKelas AS KELAS FROM
                                sis_siswa JOIN sis_kelas ON idKelasSiswa=idKelas WHERE
                                nisSiswa='$nis'");

    }

    function get_rekap_siswa_detil($nis){

        return $this->execute("SELECT namaJenisIuran AS NAMA_IURAN,
                                SUM(nominalRekap) AS NOMINAL FROM sis_rekap_iuran LEFT JOIN
                                sis_nama_iuran ON idJenisIuran=idIuranRekap WHERE
                                nisSiswaRekap='$nis'GROUP BY idiuranRekap");

    }

}
```

```

function get_list_transaksi($nis){

    return $this->execute("SELECT tanggal AS TANGGAL, CONCAT('Pembayaran ',
        namaJenisIuran) AS TRANSAKSI, nominalRekap AS TOTAL,
        idRekapSiswa AS ID_TRANS FROM sis_rekap_iuran JOIN
        sis_nama_iuran ON idIuranRekap = idJenisIuran WHERE
        nisSiswaRekap = '$nis' ORDER BY tanggal DESC");

}

}

?>

```



## Lampiran D

### Modul File Model Lihat Black List

<?

```
require_once($_SERVER['DOCUMENT_ROOT']. "/siskeu_smk2/class/mysql_db.class.php");

class blackList extends mysql_db{

    function get_black_list_uts(){

        return $this->execute("SELECT nisSiswa AS NIS,namasiswa AS NAMA,
            CONCAT(jenjangKelasSiswa, ' - ',namaKelas) AS KELAS,
            (SELECT SUM(nominalRekap) FROM sis_rekap_iuran WHERE idIuranRekap
            = 1 AND nisSiswaRekap=nisSiswa) AS TOTAL_KOMITE FROM sis_siswa
            LEFT OUTER JOIN sis_rekap_iuran ON (nisSiswa=nisSiswaRekap)
            LEFT OUTER JOIN sis_nominal_dps ON (nisSiswa=nisSiswaDps)
            LEFT JOIN sis_kelas ON (idKelasSiswa=idKelas) WHERE
            (SELECT SUM(nominalRekap) FROM sis_rekap_iuran WHERE idIuranRekap
            = 1 AND nisSiswaRekap=nisSiswa) IS NULL OR
            (SELECT SUM(nominalRekap) FROM sis_rekap_iuran WHERE idIuranRekap
            = 1 AND nisSiswaRekap=nisSiswa)
            (SELECT nominalIuran FROM sis_nominal_iuran WHERE
            idNominalJenisIuran=1 AND
            NominalJenjangKelas=jenjangKelasSiswa)*12*0.5 GROUP BY
            nisSiswa");

    }

    function get_black_list_uas(){

        return $this->execute("SELECT nisSiswa AS NIS,namasiswa AS
            NAMA,CONCAT(jenjangKelasSiswa, ' - ',namaKelas) AS KELAS,
            (SELECT SUM(nominalRekap) FROM sis_rekap_iuran WHERE
            idIuranRekap = 1 AND nisSiswaRekap=nisSiswa) AS
            TOTAL_KOMITE,(SELECT SUM(nominalRekap) FROM sis_rekap_iuran
            WHERE idIuranRekap = 2 AND nisSiswaRekap=nisSiswa) AS
            TOTAL_TA,(SELECT SUM(nominalRekap) FROM sis_rekap_iuran
            WHERE idIuranRekap = 3 AND nisSiswaRekap=nisSiswa) AS
            TOTAL_PRAKERIN,(SELECT SUM(nominalRekap) FROM
            sis_rekap_iuran WHERE idIuranRekap = 4 AND
            nisSiswaRekap=nisSiswa) AS TOTAL_OSIS,(SELECT
            SUM(nominalRekap) FROM sis_rekap_iuran WHERE idIuranRekap =
            5 AND nisSiswaRekap=nisSiswa) AS TOTAL_DPS FROM sis_siswa
            LEFT OUTER JOIN sis_rekap_iuran ON (nisSiswa=nisSiswaRekap)
            LEFT OUTER JOIN sis_nominal_dps ON (nisSiswa=nisSiswaDps)
            LEFT JOIN sis_kelas ON (idKelasSiswa=idKelas) WHERE
            (SELECT SUM(nominalRekap) FROM sis_rekap_iuran WHERE
            idIuranRekap = 1 AND nisSiswaRekap=nisSiswa) IS NULL OR
            (SELECT SUM(nominalRekap) FROM sis_rekap_iuran WHERE
            idIuranRekap = 1 AND nisSiswaRekap=nisSiswa) <
            (SELECT nominalIuran FROM sis_nominal_iuran WHERE
            idNominalJenisIuran=1 AND
            NominalJenjangKelas=jenjangKelasSiswa)*12 OR
```

```

(SELECT SUM(nominalRekap) FROM sis_rekap_iuran WHERE
idIuranRekap = 2 AND nisSiswaRekap=nisSiswa) IS NULL OR
(SELECT SUM(nominalRekap) FROM sis_rekap_iuran WHERE
idIuranRekap = 2 AND nisSiswaRekap=nisSiswa) <
(SELECT nominalIuran FROM sis_nominal_iuran WHERE
idNominalJenisIuran=2 AND
NominalJenjangKelas=jenjangKelasSiswa)*12 OR
(SELECT SUM(nominalRekap) FROM sis_rekap_iuran WHERE
idIuranRekap = 3 AND nisSiswaRekap=nisSiswa) IS NULL OR
(SELECT SUM(nominalRekap) FROM sis_rekap_iuran WHERE
idIuranRekap = 3 AND nisSiswaRekap=nisSiswa) < (SELECT
nominalIuran FROM sis_nominal_iuran WHERE
idNominalJenisIuran=3 AND
NominalJenjangKelas=jenjangKelasSiswa) OR
(SELECT SUM(nominalRekap) FROM sis_rekap_iuran WHERE
idIuranRekap = 4 AND nisSiswaRekap=nisSiswa) IS NULL OR
(SELECT SUM(nominalRekap) FROM sis_rekap_iuran WHERE
idIuranRekap = 4 AND nisSiswaRekap=nisSiswa) < (SELECT
nominalIuran FROM sis_nominal_iuran WHERE
idNominalJenisIuran=4 AND
NominalJenjangKelas=jenjangKelasSiswa) OR
(SELECT SUM(nominalRekap) FROM sis_rekap_iuran WHERE
(SELECT SUM(nominalRekap) FROM sisSiswa) IS NULL OR
sis_rekap_iuran WHERE idIuranRekap = 5 AND
nisSiswaRekap=nisSiswa) < (SELECT nominalDps FROM
sis_nominal_dps WHERE nisSiswaDps=nisSiswa)
GROUP BY nisSiswa");

```

```

}

```

```

}

```

```

?>

```