BAB 2
TINJAUAN PUSTAKA

2.1 Profil SMK Sumatra 40 Bandung

Sekolah SMK Sumatra 40 Bandung merupakan sekolah SMK yang dibangun pada bulan Juni 2008 dan merupakan bagian dari Yayasan Pendidikan R.P. HARTOYO yang sebelumnya sudah mendirikan SMP dan SMA Sumatra 40 Bandung. SMK Sumatra 40 Bandung terletak di Jl. Pahlawan No.21 Bandung. Di sekolah tersebut memiliki beberepa kompetensi keahlian yang bisa disesuaikan dengan minat dan bakat calon siswa atau siswi seperti RPL (Rekayasa Perangkat lunak) dan Akuntansi. Motto yang mereka miliki adalah Disiplin, Cerdas, dan kreatif. Para peserta didik di SMK Sumatra 40 Bandung akan diberi pembekalan yang kompeten untuk perkembangan dunia kerja.

2.1.1 Visi, Misi, dan Tujuan SMK Sumatra 40 Bandung

Visi dan misi merupakan kerangka atau tujuan utama dari suatu organisasi dengan visi merupakan tujuan jangka panjang dari suatu organisasi dan misi merupakan tujuan atau sesuatu yang harus dilaksanakan oleh suatu organisasi sebagai langkah – langkah untuk mencapai visi yang telah dirumuskan. Maka berikut penjelasan visi dan misi dari SMK Sumatra 40.

2.1.1.1 Visi

SMK Sumatra 40 Bandung diharapkan menjadi sekolah yang kreatif.

2.1.1.2 Misi

SMK Sumatra 40 Bandung melaksanakan pendidikan dan pelatihan berdasarkan kurikulum dari DEPDIKNAS, dan program ekstra kurikuler dalam rangka pengembangan bakat dan minat para siswa, sesuai dengan profesi bidang rekayasa perangkat lunak dan akuntansi, sehingga para lulusannya bisa berkompetitif pada dunia kerja.

2.1.1.3 Tujuan

Menyiapkan siswa menguasai kompetensi keahlian dan keterampilan agar siap memasuki lapangan kerja, baik sekarang maupun yang akan datang. Menyiapkan siswa agar mampu melanjutkan pendidikan ke jenjang yang lebih tinggi sesuai Dengan kejuruan. Membentuk siswa, agar memiliki jiwa

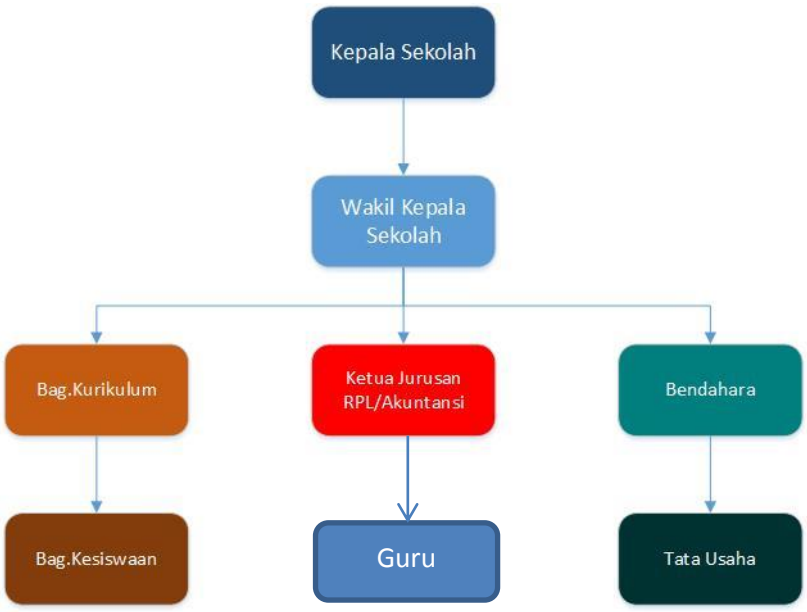
kemandirian,kewirausahaan dan kepimpinan. Membentuk siswa agar berprilaku tertib, disiplin, kreatif, berakhlak mulia, berjiwa patriotik, dan tanggap terhadap perkembangan IPTEK. Membina dan melatih kegiatan Ekstrakurikuler sesuai dengan bakat, minat, potensi siswa.

2.1.2 Struktur Organisasi

Struktur Organisasi pada gambar 2-1 merupakan sarana bagi sekolah dalam mengkoordinasikan kegiatan yang akan dilaksanakan oleh sekolah dengan melalui kerjasama antar individu untuk mencapai tujuan yang telah ditetapkan.

Dibawah ini merupakan struktur organisasi mengenai jabatan dan tugas – tugas di SMK Sumatra 40, yaitu:

1. Kepala Sekolah
2. Wakil Kepala Sekolah
3. Bagian Kurikulum
4. Bagian Kesiswaan
5. Ketua Jurusan RPL
6. Ketua Jurusan Akuntansi
7. Bendahara
8. Tata Usaha



Gambar 2-1 Struktur Organisasi SMK Sumatra 40

2.1.2.1 Deskripsi Tugas

Deskripsi tugas merupakan pemaparan, tanggung jawab dari suatu jabatan, dan berikut adalah pemaparan deskripsi tugas pada SMK Sumatra 40.

1. Kepala Sekolah
   a. Menyusun perencanaan
   b. Mengorganisasi kegiatan
   c. Mengarahkan kegiatan
   d. Mengkoordinasi kegiatan
   e. Melakukan evaluasi terhadap kegiatan
   f. Melaksanakan pengawasan
   g. Menentukan Kebijakan
   h. Mengadakan rapat
   i. Mengambil keputusan
2. Wakil kepala sekolah / Bag.Kurikulum
   a. Menyusun dan menjabarkan kalender pendidikan
   b. Menyusun pembagian tugas guru dan jadwal pelajaran
   c. Mengatur penyusunan program pengajaran, program satuan pelajaran dan persiapan pengajar, penjabaran dan penyesuaian kurikulim
   d. Mengatur pelaksanaan kegiatan ekstra kurikuler
   e. Mengatur pelaksanaan program penilaian raport
   f. Menyusun Laporan
   g. Melakukan supervise administrator dan akademis
   h. Mengatur dan mengkoordinir mata pelajaran
3. Bagian Kesiswaan
   a. Mengatur dan membina terhadap seksi OSIS
   b. Pengaturan persiapan dan pelaksanaan upacara bendera dan hari-hari besar lainnya
   c. Malaksanakan piket guna memantau prilaku/kegitan siswa siswi di Sekolah

d. Menegakan kedisiplinan siswa siswi yang meliputi ketepatan kehadiran/pemakaian seragam sekolah, dan lainnya sesuai tata tertib sekolah
e. Mengadministrasi semua kegiatan kesiswaan

4. Ketua jurusan RPL/Akuntansi
   a. Menyusun program kerja
   b. Mendalami dan mengembangkan kurikulum sesuai dengan jurusan
   c. Mengkoordinir tugas guru dan jurusan
   d. Mengkoordinir tugas kelompok kerja pengembangan kurikulum sesuai dengan jurusan
   e. Mengevaluasi analisa kurikulim dari masing masing guru sesuai dengan tugasnya
   f. Mengkoordinir penggunaan ruang praktek dalam jurusannya
   g. Membantu urusan administrasu meliputi catatan kewajiban siswa, data guru, inventarus sekolah dalam jurusannya
   h. Melaksanakan dan memelihara dengan dunia kerja/ dunia industry data instansi yang terkait
   i. Pelaksanaan bimbingan penyuluhan kejuruan dalam jurusannya
   j. Supervisi dan evaluasi kegiatan belajar mengajar dan tugas lain dalam jurusannya
5. Bendahara
   a. Menerima uang, membukukan uang masuk dan keluar
   b. Membayar uang gaji guru dan pegawai
   c. Membayar uang belanja sekolah
   d. Mengeluarkan uang sesuai dengan APBS atas persetujuan kepala sekolah
   e. Menyimpan arsip keuangan sekolah

6. Tata Usaha
    a. Membantu mengatur pengurusan kepegawaian
    b. Meneliti dan kemudian membuat surat, baik surat masuk maupun surat keluar sesuai dengan disposisi/instruksi kepala sekolah
    c. Mengawasi dan mengendalikan penggunaan alat – alat sekolah
    d. Membantu Kepala sekolah dalam penyusunan RAPBS
    e. Bertanggung jawab atas penggunaan stample sekolah
    f. Mengurus/mengerjakan buku induk siswa
    g. Menyiaapkan dan mengisi buku klaper
    h. Mengurus presensi siswa da jurnal kelas
    i. Mengerjakan lager nilai
    j. Membuat data static dan rekaptulasi siswa tiap bulan
    k. Mengelola administrasi beasiswa
    l. Menangani pengarsipan dokumen kesiswaan
    m. Membantu dan melaksanakan tugas lain yang relevan yang diberikan atasan langsung

2.2 Landasan Teori

Landasan teori adalah kumpulan dari definisi dan konsep dari berbagai sumber. Landasan teori ini menjadi dasar yang kuat dalam sebuah penelitian, maka beberapa landasan teori yang dijelaskan adalah landasan teori yang menunjang untuk pengembangan layanan SMS Gateway pada sistem informasi akademik di SMK Sumatra 40 Bandung.

2.2.1 Pengembangan

Pengembangan adalah suatu usaha untuk meningkatkan kemampuan teknis, teoritis, konseptual, dan moral sesuai dengan kebutuhan. Ada berbagai macam pengembangan pendidikan , pengembangan sistem secara umum dapat diberikan dua buah model pengembangan yang baru yaitu: Pertama "*top-down model*" yaitu pengembangan pendidikan yang diciptakan oleh pihak tertentu sebagai pimpinan/atasan yang diterapkan kepada bawahan; seperti halnya pengembangan

pendidikan yang dilakukan oleh Departemen Pendidikan Nasional selama ini "*bottom-up model*" yaitu model pengembangan yang bersumber dan hasil ciptaan dari bawah dan dilaksanakan sebagai upaya untuk meningkatkan penyelenggaraan dan mutu pendidikan. Kemudian pengembangan sistem, antara lain : perlunya pengembangan sistem, prinsip-prinsip dalam pengembangan sistem, siklus hidup pengembangan sistem, dan pendekatan pengembangan sistem itu sendiri, perbedaan antara metodologi, metode dan algoritma, beberapa klasifikasi dari siklus hidup pengembangan sistem, teknik dalam pengembangan sistem dan perbedaan antara sistem analis dan programer.

Perlunya Pengembangan Sistem dengan seiringnya perkembangan jaman maka sebuah sistem tentu tidak selamanya dapat digunakan dengan baik. Untuk itu perlu ada perubahan terhadap sistem tersebut baik dengan cara memperbaiki sistem yang lama ataupun jika perlu untuk mengganti sistem yang lama. Ada beberapa hal yang mendasari hal tersebut, antara lain:

Ada permasalahan pada sistem yang lama Permasalahan yang dimaksud ini seperti adanya ketidak beresan pada sistem yang lama sehingga hasilnyapun tidak sesuai dengan yang diharapkan. Contohnya : terdapat kesalahan-kesalahan baik yang disengaja ataupun tidak yang menyebabkan data pada suatu perusahaan tidak dapat terjamin kebenarannya, adanya kesempatan atau peluang anggota dari sistem tersebut untuk melakukan kecurangan. Permasalahan yang lain juga dapat disebabkan oleh pertumbuhan organisasi tersebut.

Untuk meraih kesempatan (*opportunities*) Sebuah sistem harus diperbaiki atau dikembangkan juga disebabkan untuk meraih kesempatan dari suatu organisasi atau perusahaan.

Adanya intruksi-intruksi (*directives)* Sistem harus diperbaharui atau dikembangkan juga disebabkan oleh faktor eksternal seperti pemerintah. Adanya kebijakan-kebijakan pemerintah memaksa sebuah perusahaan menggunakan sistem yang tidak bertentangan dengan kebijakan tersebut.

Pengembangan atau pembuatan sebuah sistem tentu tidak memakan biaya yang sedikit, sehingga organisasi harus secara bijak menentukan apakah sistem yang

digunakan masih layak untuk dipakai atau sudah harus dikembangkan atau diganti.

Tujuan Pengembangan Sistem :

1. *Performance* (kinerja), dapat diukur dengan 2 parameter yaitu *throughput* dan *respon time. Throughput* adalah banyaknya transaksi yang dapat dilakukan pada suatu saat tertentu. Respon time adalah waktu yang terbuang pada saat perpindahan proses transaksi. Jadi peningkatan kenerja bertujuan untuk meningkatkan jumlah transaksi dengan waktu yang secepat mungkin.
2. *Information* (Informasi), peningkatan kualitas dari informasi tersebut. Tentu saja kualitas dari informasi ini akan menentukan kebijakan dari organisasi tersebut.
3. *Economy*, meningkatkan keuntungan dengan biaya yang minimum.
4. *Control* (pengendalian), Digunakan untuk mengontrol atau mendeteksi adanya kesalahan-kesalahan pada suatu sistem.
5. *Efficiency* (efisiensi), berbeda dengan ekonomis yang bertitik berat pada keuangan, efisiensi disini adalah pemanfaatan sumber daya semaksimal mungkin.
6. *Service*, peningkatan layanan oleh sebuah sistem.

Pengembangan ini dilakukan di SMK Sumatra 40 pada SI Kesiswaan yang sudah ada, SI Kesiswaan dikelola oleh ketua jurusan RPL .

2.2.2 Pengertian Sistem

Sistem berasal dari bahasa Yunani “Systema” yang memiliki pengertian yang luas, tidak ada satu definisi yang pasti. Terdapat dua kelompok pendekatan di dalam pendefenisian sistem, yaitu yang menekan pada prosedur dan menekan pada komponen atau elemennya. Pendekatan pada sistem yang lebih menekankan pada prosedur, mendefenisikan sistem sebagai berikut:

“Suatu sistem adalah jaringan kerja dari prosedur – prosedur yang saling berhubungan, berkumpul bersama – sama untuk melakukan suatukegiatan atau untuk menyelesaikan suatu sasaran tertentu”.

Pendekatan sistem yang lebih menekankan pada elemen atau komponennya, mendefenisikan sebagai berikut:

“Sistem adalah kumpulan – kumpulan dari elemen – elemen yang berinteraksi untuk mencapai tujuan tertentu”.

Berdasarkan pengertian sistem yang dijelaskan sebelumnya, maka sistem adalah kumpulan elemen – elemen atau prosedur yang saling berhubungan dan bertanggung jawab memproses data masukan (*input*) sehingga menghasilkan suatu keluaran (*output*). Bagian listing yang paling kecil disebut subsistem, misalnya sistem komputer yang terdiri dari bagian subsistem perangkat keras dan subsistem perangkat lunak. Masing – masing subsistem dapat terdiri dari subsistem – subsitem atau komponen – komponen yang lebih kecil lagi, komponen – komponen ini membentuk satu kesatuan sehingga tujuan atau sasaran dari sistem tersebut dapat tercapai.

Tujuan sistem adalah untuk menentukan operasi yang dilaksanakan. Berkembangnya suatu sistem dapat dipengaruhi oleh informasi yang terdapat di dalamnnya. Jika berkurangnya informasi, maka suatu sistem lama kelamaan akan berakhir dan tidak dapat digunakan lagi.

Sistem yang akan ditambahkan dipenelitian ini adalah sistem layanan SMS Gateway pada SI Kesiswaan,

2.2.3 Pengertian Data dan Informasi

Data dan informasi mempunyai keterkaitan antara satu dengan yang lainnya. Data adalah fakta – fakta atau pesan – pesan yang belum dievaluasi atau bahan mentah dari informasi yang melalui pengolahan tertentu dibentuk menjadi suatu informasi. Data merupakan kenyataan yang menggambarkan suatu kejadian – kejadian nyata.

Suatu sistem yang kurang mendapatkan suatu informasi akan luruh, kerdil, dan akhirnya berakhir. Defenisi informasi sendiri dijelaskan pada uraian berikut:
“Informasi adalah data yang diolah menjadi bentuk yang lebih berharga dan lebih berguna, serta lebih berarti bagi yang menerimanya”.
Sedangkan sumber dari informasi sendiri adalah data yang berarti kenyataan untuk menggambarkan suatu kejadian – kejadian dan kesatuan yang nyata. Maksud dari kejadian – kejadian tersebut adalah sesuatu yang terjadi pada saat tertentu

Suatu sistem merupakan suatu keseluruhan yang bulat dan utuh, dimana tujuan dari masing – masing bagian yang membentuk sistem akan saling menunjang dan mencapai tujuan dari suatu sistem secara keseluruhan. Maka dari itu, tujuan yang dicapai dari salah satu bagian tidak dapat mengabaikan pencapaian tujuan dari bagian yang lain.

Sumber dari informasi adalah data. Data terbentuk dari karakter – karakter yang dapat berupa alphabet, angka maupun simbol khusus. Data disusun untuk diolah dalam bentuk struktur data, struktur file dan database. Terdapat perbedaan antara data dan informasi, sedangkan informasi digunakan dalam pengambilan keputusan, karena itu informasi memiliki tingkat lebih tinggi dari data.

Data informasi yang dibutuhkan untuk layanan SMS Gateway adalah data nilai , data absensi ,dan data SPP kemudian dihubungkan dengan data siswa , data guru, data orang tua.

2.2.4 Pengertian Sistem Informasi

Sistem informasi berfungsi untuk menyediakan informasi yang berguna untuk manajemen yang mendukung rencana strategis organisasi. Sedangkan tujuannya adalah untuk mengumpulkan data, menyimpan data dan menginformasikannya kepada para pemakai.

Menurut **Robert A. Leitc dan K. RescoeDavis**, Sistem Informasi didefinisikan sebagai berikut :

“Sistem informasi adalah suatu sistem di dalam organisasi yang mempertemukan kebutuhan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi yang menyediakan bagi pihak luar tertentu dengan laporan yang diperlukan”.

Sistem informasi dapat berjalan apabila ada elemen – elemen berikut:

a. Perangkat keras (*Hardware*), yaitu komputer yang berperan sebagai media masukan, proses dan keluaran.
b. Perangkat lunak (*Software*), yaitu alat yang digunakan untuk menjalankan perangkat keras yang dapat berupa sistem operasi atau program aplikasi.

c. Pengguna komputer (*Brainware*), adalah manusia yang merupakan bagian terpenting yang dapat menangani semua elemen penggerak dari suatu sistem informasi.
d. Data, yaitu fakta – fakta dari suatu kejadian yang dapat diolah untuk menghasilkan suatu informasi.

Prosedur, yaitu urutan kerja secara sistematis agar suatu pekerjaan dapat dilaksanakan secara teratur sesuai dengan rencana.

Sistem Informasi yang ada saat ini adalah Sistem Informasi Kesiswaan berupa informasi data nilai, data absensi , dan data pembayaran.

2.2.5 Pengertian SMS (*Short Message Service*)

Short Message Service (SMS) adalah kemampuan untuk mengirim dan menerima pesan dalam bentuk teks dari dan kepada ponsel. Teks tersebut bisa terdiri dari kata-kata atau nomor atau kombinasi alphanumeric.

SMS diciptakan sebagai standar pesan (*message*) oleh ETSI (*Europesan Telecommunication Standards Institute*), yang juga membuat standar GSM (*Global System for Mobile*) yang diimplementasikan oleh semua operator *Global System for Mobile* (GSM). SMS yang pertama dikirimkan pada Desember 1992 dari PC ke sebuah ponsel melalui jaringan *Global System for Mobile* (GSM) Vodafone di UK. Setiap Pesan maksimal terdiri dari 160 karakter jika menggunakan alphabet Latin, dan 70 karakter jika menggunakan alphabet non-Latin seperti huruf Arab atau China.

Pemanfaatan SMS saat ini sangat popular dibandingkan yang lainnya, disamping tarifnya murah tetapi juga mudah dan cepat dalam pemakaiannya sehingga sangat digemari oleh siapapun. Saat ini SMS juga mulai digunakan sebagai media penyebaran pesan ke banyak pengguna (*broadcast*) yang idenya diadopsi sebagai fasilitas peringatan dini (*early warning system*) yang digunakan sebagai peringatan akan kemungkinan terjadinya suattu bencana. SMS juga mulai dikembangkan sebagai perangkat pendaftaran elektronik (*digital sign*), perangkat penjawab umpan balik otomatis (*auto responder*), dan SMS dapat difungsikan sebagai alat *control* aktivasi (*remote controller activator*) ke dalam perangkat

computer seperti untuk mematikan computer, menyalakan lampu, menyalakan sistem keamanan, mengetahui posisi dan lain-lain. Secara umum SMS telah dikembangkan untuk (**Zakaria & Josef**, 9) :

a. Sistem pemilihan (*polling*) dan jajak pendapat (*voting*).
b. Pengiriman banyak serentak (*broadcasting*)
c. Pengumuman (*announcer*)
d. Peringatan (*warning*)
e. Pengingat (*reminder*)
f. Alat *control* aktivasi (*remote controller activator*)
g. Penjawab umpan balik otomatis (*auto responder*)
h. informasi (*Information*)
i. Pendaftaram bertingkat (*cascading digital sign*)
j. Pendaftaran tersusun (*smart tag digital sign*)
k. Perangkat penguji jaringan (*tester*)
l. *Flood* SMS/SMS *Bomb*
m. *Floating* SMS, dan lain-lain.

Menurut riswan (2006), Teknologi SMS memiliki banyak keunggulan keunggulan tersebut antara lain :

a. Harga untuk mengirim sebuah SMS murah
b. Merupakan “deliver oriented service”, artinya pesan akan selalu diusahakan untuk dikirimkan ke nomor tujuan. Jika suatu saat nomor tujuan sedang aktif atau diluar cakupan jaringan, maka pesan akan disimpan di *SMSC server* dan akan dikirimkan sesegera setelah nomor tujuan aktif kembali. Pesan juga akan tetap terkirim ke tujuan walaupun nomor tujuan sedang melakukan pembicaraan (sibuk).
c. Dapat dikirim ke banyak penerima sekaligus pada saat yang bersamaan.
d. Pesan dapat dikirimkan ke berbagai jenis tujuan, seperti e-mail, Ip ataupun aplikasi lain.

Dari beberapa pernyataan diatas dapat diketahui bahwa SMS ini memiliki banyak manfaat bagi sistem yang saat ini akan dibangun , yang digunakan dalam sistem ini adalah *broadcast* dan *auto respond* , dua metode tersebut digunakan di

layanan SMS Gateway ini untuk mengirim informasi berupa data nilai , SPP , absen dan info sekolah untuk masing masing – siswa dan orang tua siswa.

2.2.6 layanan SMS

"layanan aplikasi pada SMS pada dasarnya memiliki karakteristik yang berbeda dengan aplikasi internet yaitu ukuran layar monitor yang berukuran kecil, serta keterbatasan jumlah karakter yang bisa dikirimkan". **Zakaria & Josef** (2006 : 34)

Dari karakteristik tersebut dapat mendasari pada pengembangan aplikasi yang dapat menyediakan informasi secara singkat dan jelas dengan pengoperasian aplikasi yang mudah dan sederhana ini. Berdasarkan mekanisme distribusi pesan SMS oleh aplikasi layanan SMS, terdapat 2(dua) macam mekanisme penghantaran pesan,

yaitu :

a. *Pull*, yaitu pesan yang dikirimkan ke pengguna berdasarkan permintaan pengguna, dilayanan SMS Gateway ini digunakan ketika permintaan di kirim kemudian akan merespon secara otomatis.
b. *Push-Event based*, yaitu pesan yang diaktivasi oleh aplikasi berdasarkan kejadian atau kondisi yang akan berlangsung, kondisi yang berlangsung dilayanan SMS Gateway ini adalah ketika Nilai sudah dimasukan oleh guru, absensi dimasukan oleh walikelas dan sudah melakukan transaksi SPP.

2.2.7 SMS *Gateway*

"Istilah *gateway*, bila dilihat pada kamus Inggris-Indonesia diartikan sebagai pintu gerbang. Namun pada dunia komputer, *gateway* dapat berarti juga sebagai jembatan penghubung antara satu sistem dengan sistem lain yang berbeda, sehingga dapat terjadi suatu pertukaran data antara sistem tersebut. Dengan demikian, SMS *gateway* dapat diartikan sebagai suatu penghubung untuk lalulintas data-data SMS, baik yang dikirimkan maupun yang diterima."

Yunianto (2006:25)

Ada dua cara mengirimkan pesan dari kompter / *mobile phone*:

1. Sambungkan *mobile phone* atau GSM/GPRS *modem* ke kompter/PC. Kemudian gunakan komputer/PC dan *AT Commands* untuk

menginstruksikan *mobile phone* atau GSM/GPRS modem untuk mengirimkan pesan sms.

2. Sambungkan komputer/PC ke sms *center* (SMSC) atau sms gateway dari perusahaan *wireless* atau sms *service provider*. Lalu kirimkan pesan sms dengan menggunakan protokol yang disupport oleh SMSC atau sms *gatewa*y.

Ada tiga cara untuk menerima pesan sms dengan menggunakan komputer / pc:

1. Sambungkan mobile phone atau GSM/GPRS modem ke komputer/pc. Kemudian gunakan komputer / pc dan *AT Commands* untuk menerima sms dari *mobile phone* atau GSM/GPRS modem.
2. Dapatkan akses ke sms *center* (SMSC) atau sms gateway dari perusahaan wireless. Pesan sms yang diterima akan dikirimkan ke komputer / pc dengan menggunakan protokol yang disupport oleh SMSC atau sms gateway.
3. Dapatkan akses ke sms *gateway* dari sms *service provider*. Pesan sms yang diterima akan dikirimkan ke komputer / pc dengan menggunakan protokol yang di-*support* oleh sms *gateway*.

"Sistem minimal dari SMS *Gateway* adalah sebuah *Personal Computer* yang dilengkapi dengan aplikasi SMS *Gateway*, perangkat komunikasi berbasis *selluler* (Telpon Selluler atau Modem GSM/CDMA) yang terhubung dengan PC. Aplikasi SMS Gateway adalah sebuah perangkat lunak yang mampu membaca informasi SMS yang diterima dan melakukan pengelolaan sesuai kebutuhan."
Dewanto (2007:17)

Untuk koneksi dengan telpon selluler menggunakan perintah AT Command melalui *port I/O* (*serial/USB*) yang mendukung GSM Modem ataupun berbagai merk telpon selluler. Untuk aplikasi pembacaan data SMS menggunakan konversi PDU Format dan metoda parsing untuk memisahkan kode-kode transaksi. Pembacaan dan pengelolaan pesan berikutnya menggunkan sebuah aplikasi berbasis Sistem Informasi Manajemen.

Istilah *gateway* dapat diartikan sebagai pintu gerbang. Namun pada dunia komputer, *gateway* dapat diartikan sebagai jembatan penghubung antara satu

sistem dengan sistem yang lain, sehingga dapat terjadi pertukaran data antar sistem tersebut. Dengan demikian, SMS *Gateway* dapat diartikan sebagai pintu gerbang atau jalur bagi penyebaran informasi dengan menggunakan *Short Message Service* (SMS) Dengannya, kita dapat menyebarkan pesan ke banyak nomor sekaligus secara otomatis dan cepat. Dengan menggunakan program tambahan yang dapat dibuat sendiri, si pengirim pesan dapat lebih fleksibel dalam mengirim/menerima *Short Message Service* (SMS) sesuai dengan aturan yang ditetapkan, seperti sistem untuk *request and reply* di SMK Sumatra 40 yang akan dikembangkan ini misalnya. Cara kerja SMS *Gateway* : SMS dikirim orang tua ke nomor yang telah ditentukan, isi SMS merupakan Format yang telah ditentukan sebelumnya, Cara kerja SMS Gateway dipenelitian ini atau SMS diterima oleh *Modem* yang bekerja sebagai SMS *Gateway* kemudian akan dikirimkan ke sistem Layanan SMS Gateway kemudian sistem akan merespon SMS tersebut. Jika Format SMS benar maka *Modem* yang bertindak sebagai SMS *Gateway Server* akan membalas begitu juga sebaliknya. Kemudian SMS tersebut ditampung dalam database dan akan ditampilkan kedalam Website layanan SMS Gateway.

2.2.8 *Codeignitor*

Codeignitor Saat ini *Framework* PHP yang populer digunakan. Tidak hanya di Indonesia saja, tetapi di luar negeri sana juga sangat pupuler. Semakin banyak perusahaan yang mencari calon *programmer* yang mahir menggunakan *CodeIgniter* dalam membangun aplikasi berbasis *website.*

CodeIgniter adalah aplikasi open source yang berupa *framework* dengan model MVC (*Model, View, Controller*) untuk membangun *website* dinamis dengan menggunakan PHP. *CodeIgniter* memudahkan *developer* untuk membuat aplikasi *webiste* dengan cepat dan mudah dibandingkan dengan membuatnya dari awal.

CodeIgniter dirilis pertama kali pada 28 Februari 2006.

Model View Controller merupakan suatu konsep yang cukup popular dalam pembangunan aplikasi web, berawal pada bahasa pemrograman *Small Talk*,

MVC memisahkan pengembangan aplikasi berdasarkan komponen utama yang membangun sebuah aplikasi seperti manipulasi data, *user interface*, dan bagian

yang menjadi kontrol aplikasi. Terdapat 3 jenis komponen yang membangun suatu MVC *pattern* dalam suatu aplikasi yaitu :

1. *View*, merupakan bagian yang menangani *presentation logic*. Pada suatu aplikasi *web* bagian ini biasanya berupa *file template* HTML, yang diatur oleh *controller*. *View* berfungsi untuk menerima dan merepresentasikan data kepada user. Bagian ini tidak memiliki akses langsung terhadap bagian model.
2. Model, biasanya berhubungan langsung dengan *databases* untuk memanipulasi data (*insert, update, delete, search*), menangani *validasi* dari bagian *controller*, namun tidak dapat berhubungan langsung dengan bagian *view*.
3. *Controller*, merupakan bagian yang mengatur hubungan antara bagian model dan bagian *view*, controller berfungsi untuk menerima request dan data dari *user* kemudian menentukan apa yang akan diproses oleh aplikasi.

Ada beberapa kelebihan *CodeIgniter* (CI) dibandingkan dengan *Framework* PHP lain :

1. Performa sangat cepat. Salah satu alasan tidak menggunakan *framework* adalah karena eksekusinya yang lebih lambat daripada PHP *from the scracth*, tapi *CodeIgniter* sangat cepat bahkan mungkin bisa dibilang *CodeIgniter* merupakan *framework* yang paling cepat dibanding *framework* yang lain.
2. Konfigurasi yang sangat minim (*nearly zero configuration*). Tentu saja untuk menyesuaikan dengan *databases* dan keleluasaan *routing* tetap diizinkan melakukan konfigurasi dengan mengubah beberapa *file* konfigurasi seperti database.PHP atau autoload. PHP, namun untuk menggunakan *CodeIgniter* dengan *setting standard*, anda hanya perlu merubah sedikit saja *file* pada *folder config*.
3. *CodeIgniter* memiliki banyak komunitas. Dengan banyaknya komunitas CI ini, memudahkan kita untuk berinteraksi dengan yang lain, baik itu bertanyaatau teknologi terbaru.

4. Dokumentasi yang sangat lengkap. Setiap paket instalasi *CodeIgniter* sudah disertai *user guide* yang sangat bagus dan lengkap untuk dijadikan permulaan, bahasanya pun mudah dipahami.
5. Tutorial sederhana untuk membantu membuat sistem layanan SMS Gateway cukup banyak dan mudah didapat.

Codeignitor adalah framework yang digunakan untuk mengembangkan layanan SMS Gateway di SMK Sumatra 40 Bandung dengan bahasa pemprograman PHP.

2.2.9 Gammu

Gammu merupakan salah satu pustaka atau *library open-source* yang dibuat sebagai *gateway* antara *handphone* dengan perangakat komputer. Pengembangan Gammu awalnya dari pendahuluannya yaitu Gnooki yang dari segi konsep masih sangat sederhana dan rumit. Namun pada Gammu proses instalasi dan penerapan pembangunan aplikasi SMS semakin mudah. Ini merupakan jasa dari beberapa *developer* yang telah membangun Gammu.

Gammu adalah semacam *service* yang disediakan untuk membangun aplikasi yang berbasis SMS *Gateway*. Setelah sukses membangun s*ervice* Gammu, maka bisa menggunakan aplikasi Layanan SMS *Gateway* data-data SMS akan diterima oleh Gammu ini kemudian diproses oleh Sistem.

Kelebihan dari Gammu ini adalah merupakan *software open source* yang tersedia dalam bentuk *source code*. Selain itu Gammu juga tersedia untuk *Operating Sistem* Windows.

Gammu dalam penelitian ini digunakan sebagai service SMS untuk menyimpan data SMS dan juga untuk mengirim dan menerima semua data sms yang akan dikirim kepada orang tua.

2.2.10 Apache

Apache adalah sebuah HTTP server. Apache dibangun pertama kalinya berdasarkan pada kode-kode dan ide-ide yang terdapat pada HTTP *server* yang terkenal yaitu NCSA httpd 1.3, pada awal 1995.

Pada saat ini Apache termasuk web *server* yang paling banyak digunakan, hal ini terjadi bukan semata-mata karena Apache diedarkan secara gratis. Namun

juga karena kemampuannya, secara fungsional, tingkat efisien dan kecepatannya yang dapat dibandingkan dengan *server* HTTP UNIX lainnya. Selain diedarkan secara gratis dan *open source*, pengembangan Apache juga dilakukan oleh banyak *programmer* secara sederhana, Apache digunakan untuk kebutuhan *Localhost server* untuk menjalankan aplikasi Layanan SMS Gateway.

2.2.11 MySql

Menurut **Madcom** (2004 : 177) MySQL merupakan *software database* yang ternasuk paling populer di lingkungan Linux, kepopuleran ini karena ditunjang performansi *query* dari databasenya yang saat itu bisan dikatakan paling cepat dan jarang bermasalah.Berangkat dari *software* yang *shareware* MySQL populer, kini mulai dari versi 3.23 MySQL menjadi *software open source* yang berarti *free*. MySQL dapat digunakan untuk kepentingan komersial atau pun personal (non profit).

MySQL telah tersedia juga di lingkungan windows, *software* MySQL di lingkungan Windows dipasang pada direktori c:\mysql. c:\mysql\bin adalah direktori yang berisi daftar modul *executable* dari *software* MySql. PHP untuk Windows secara *default* telah mendukung MySQL.

MySQL ini diguanakan untuk menyimpan Data yang dibutuhkan untuk layanan SMS Gateway di SMK Sumatra 40, seperti data nilai, data absen , dan data SPP.

2.2.12 PHP (PHP *Hypertext Preprocessor*)

PHP adalah *script* untuk pemrograman *web*,PHP ini digunakan untuk membangun layanan SMS Gateway ini kemudian PHP dapat digunakan bersama – sama dengan kode HTML (*embedded*). Sehingga hampir mirip dengan javascript, bedanya dengan javascript adalah javascript akan diterjemahkan oleh *internet client* (*browser*), sedangkan *script* PHP diterjemahkan di pihak *server*. Karenanya PHP juga dikenal sebagai *server side script*. Halaman hasil proses di *server*, diterima oleh client internet sebagai HTML biasa, dan kode – kode PHP tidak akan bisa terbaca oleh *client*. Hal ini juga yang membedakan PHP dengan javascript.

Script PHP ini mulai bekerja saat adanya *request* PHP file tersebut dari *Web browser*. *Web server* kemudian akan memanggil *file* tersebut, dan akan membaca semua *request* dari atas sampai bawah. Mengeksekusi beberapa *script command*, dan mengirimkan hasilnya dalam sebuah halaman HTML melalui *browser*.

PHP merupakan bahasa pemrograman yang digunakan dalam pengembangan layanan SMS Gateway pada SI Kesiswaan ini.

2.2.13 HTML (*HyperText Markup Language*)

HTML adalah suatu bahasa pemrograman yang dipergunakan untuk menyusun dan membentuk suatu dokumen agar dapat ditampilkan pada program *browser* WWW (*world wide web*) dalam bentuk yang dikehendaki oleh pembuatnya di *codeignitor* HTML merupakan View atau tampilan ketika *script* dipanggil.

HTML juga dipergunakan untuk menciptakan *hypertext link* atau hubungan antara teks dan dokumen lain atau juga bagian dari dokumen lain, HTML hanya berupa file teks yang bisa dibaca dengan suatu editor teks biasa, sesuai dengan namanya bahasa ini menggunakan tanda (*markup*) untuk menandai perintah perintahnya.

HTML dalam layanan SMS Gateway ini merupakan tampilan atau view yang nantinya akan dipanggil oleh PHP, HTML ini sebagai *function*.

2.3 Berorientasi Objek

Menurut **Adi Nugroho** (2005 : 4) yang dimaksud dengan pendekatan berorientasi objek adalah. Suatu cara baru dalam berpikir serta berlogika dalam menghadapi masalah-masalah yang akan dicoba-atasi dengan bantuan komputer, mencoba melihat permasalahan lewat pengamatan dunia nyata dimana setiap objek adalah entitas tunggal yang memiliki kombinasi struktur data dan fungsi tertentu, seperti entitas Siswa , guru , walikelas dan lainnya untuk mendukung pengembangan layanan SMS Gateway.

Berorientasi Objek merupakan Suatu cara pengembangan perangkat lunak dan sistem informasi berdasarkan objek-objek yang ada di dunia nyata. Teknologi berorientasi objek menganalogikan sistem aplikasi seperti kehidupan nyata yang

didominasi objek. Berikut prinsip dasar dari pemograman berorientasi objek yang terdiri dari :

1. Abstraksi : memfokuskan pada karakteristik objek.
2. Kelas : bagian yang dapat menciptakan objek
3. Enkapsulasi (Pembungkusan) : Menyembunyikan banyak hal yang terdapat dalam objek yang tidak perlu diketahui objek lain
4. Generlaisasi (Spesialisasi) : pembagian kelas-kelas yang berperilaku sama.
5. Polimorfisme : fungsi yang sama dapat diterapkan dan dimiliki oleh kelas-kelas berlainan.

OOP didalam penelitian ini digunakan dalam pembangunanan sistem atau pengembangannya yang berorientasi objek.

2.3.1 UML (*Unified Modeling Language*)

UML dikeluarkan oleh OMG (*Object Management Group, Inc)* yaitu organisasi internasional yang dibentuk pada 1989, terdiri dari perusahaan sistem informasi, *software developer* dan para *user* sistem komputer.

UML ini membantu untuk menggambarkan bagaimana layanan SMS Gateway dapat diterapkan di SI Kesiswaan , beberapa konsep dasar dalam OOP (*Object Oriented Programming*) menurut **Martin Flower** :

1. Object

Objek (*object*) adalah "benda", secara fisik atau konseptual, yang dapat kita temui disekeliling kita. *Hardware*, *software*, dokumen, manusia, dan bahkan konsep semuanya adalah contoh objek. Untuk kepentingan memodelkan perusahaannya seperti, kepala sekolah, guru, wali kelas , siswa, dokumen, merupakan keuntungan sebagai objek.

Sebuah objek memiliki keadaan sesaat (*state*) dan perilaku (*behavior*). State dari sebuah objek adalah kondisi objek tersebut atau himpunan dari keadaan yang menggambarkan objek tersebut.

Sebagai contoh, walikelas adalah objek, dan salah satu keadaan sudah memasukan absensi atau tidak adalah state dari objek wali kelas tersebut. Salah satu state dari objek jam adalah waktu saat ini. State dinyatakan dengan nilai dari atribut (*attribute*) objeknya. Atribut adalah nilai internal suatu objek yang

mencerminkan antara lain karakteristik objek, kondisi sesaat, koneksi dengan objek lain, dan identitasnya. Perubahan state dicerminkan oleh perilaku (*behavior*) objek tersebut. Sebagai contoh, jika pengisian nilai merupakan perilaku (behavior) atau *service*-nya mengisi nilai apapun yang dilakukan oleh siswanya.

Behavior suatu objek mendefinisikan bagaimana sebuah objek bertindak (beraksi) dan memberi reaksi. *Behavior* ditentukan oleh himpunan semua atau beberapa operasi yang dapat dilakukan dalam objek itu sendiri. *Behavior* dari sebuah objek dicerminkan oleh *interface*, *service*, dan *method* dari objek tersebut. *Interface* adalah pintu untuk mengakses *service* objek. *Service* adalah fungsi yang bisa diemban objek. Method adalah mekanisme internal objek yang mencerminkan perilaku (*behavior*) objek tersebut. Sebagai contoh, jika printer sebuah merupakan perilaku (*behavior*) atau service-nya mencetak apapun yang dia terima.

2. Class

Kelas (*class*) adalah definisi umum (pola, *template* atau cetak biru) untuk himpunan objek sejenis. Kelas menetapkan spesifikasi perilaku (*behaviors*) dan atribut objek-objek tersebut. *Class* adalah keniskalan (abstraksi) dari entitas dalam dunia nyata. Objek adalah "contoh" (*instance*) dari sebuah kelas. Sebagai contoh, atribut untuk kelas *autoreply* adalah saat mendapatkan SMS masuk yang sesuai dengan format, Perilakunya adalah format yang sudah ditentukan maka akan otomatis membalas SMS masuk terebut. Contoh (*instance*) yang mungkin dari kelas autoreply ini adalah SMS masuk, SMS keluar.

Inheritance

Seperti yang sudah diuraikan di atas, objek adalah contoh/ *instance* dari sebuah *class*. Hal ini mempunyai konsekuensi yang penting yaitu sebagai *instande* sebuah *class*, sebuah objek mempunyai semua karakteristik dari *class*-nya. Inilah yang disebut dengan inheritance (pewarisan sifat). Dengan demikian apaun *attribute* dan *operation* dari *class* akan dimiliki pula oleh semua objek yang diinherit/ diturunkan dari class tersebut. Sifat ini tidak

hanya berlaku untuk objek terhadap *class*, akan tetapi juga berlaku untuk *class* terhadap *class* lainnya.

3. *Polimorphisme*

Polimorphisme adalah konsep yang handal bagi pengembangan perangkat lunak untuk pemisahan secara jelas diantara sub sistem yang berbeda. Dengan demikian sebuah sistem akan bisa dimodifikasi secara mudah karena hanya dibutuhkan *interface* antar *class*.

4. *Encapsulation*

Encapsulation adalah proses menyembunyikan detil implementasi sebuah objek. Satu-satunya jalan untuk mengakses data objek tersebut adalah melalui *interface*. *Interface* melindungi internal state sebuah objek dari "campur tangan" pihak luar. Oleh karena itu objek sering digambarkan sebagai kotak hitam (*black box)* yang menerima dan mengirim pesanpesan (*messages*). Dalam *object-oriented programming* kotak hitam tersebut berisi kode (himpunan intruksi dengan bahasa yang dipahami komputer) dan data (informasi dimana intruksi tersebut beroperasi dengannya).

Dalam *object-oriented programming*, kode dan data disatukan dalam sebuah "benda" yang tersembunyi isinya, yaitu objek. Pengguna objek tidak perlu mengetahui isi dalam kotak tersebut. Untuk dapat berkomunikasi dengan objek, diperlukan pesan (*messages*).

Secara formal kita definisikan message sebagai permintaan untuk objek penerima (*receiver object*) untuk membawa metode yang ditunjukkan atau perilaku dan mengembalikan result dari aksi tersebut kepada objek pengirim (*sender object*). Sebagai contoh, satu object orang mengirim object bola lampu sebuah pesan (*message*) untuk menyala (melalui saklar). Objek bola lampu menyalakan dirinya dan menunjukkan kepada objek orang tersebut bahwa state barunya adalah menyala.

5. *Association dan Aggregation*

Association (asosiasi) adalah hubungan antar objek yang saling membutuhkan. Sedangkan, *aggregation* (agregasi) adalah bentuk khusus dari asosiasi yang menggambarkan seluruh bagian suatu objek merupakan bagian dari

objek lainnya. Sebagai contoh, objek tanggal dapat disusun dari objek hari, objek bulan, dan objek tahun. Satu alasan mengapa *objectoriented* menguntungkan bagi *programmer* adalah karena programmer dapat mendesain program dalam bentuk objek-objek dan hubungan antar objek tersebut untuk kemudian dimodelkan dalam sistem nyata.

Keuntungan yang lain adalah proses pembuatan *software* dapat dilakukan dengan lebih cepat karena *software* dibangun dari *object-object* standar yang ada di SMK Sumatra 40, kemudian dapat menggunakan model yang ada, dan dapat membuat model dengan cepat melalui metodologi. Kualitas yang tinggi dari *software* dapat dicapai karena adanya *tested components*. Lebih mudah dalam maintenance karena perbaikan kode hanya diperlukan pada satu tempat (bukan diurut dari awal). Mudah dalam membangun sistem yang besar karena subsistem dapat dibuat dan diuji secara terpisah. Mengubah sistem yang sudah ada tidak memerlukan membangun ulang keseluruhan sistem.

UML sendiri dipenelitian ini merupaka suatu alat untuk mengambarkan kerja atau flow pembangunan SMS Gateway ini.

2.3.1.1 Alat Bantu UML

Alat bantu UML sebagaimana yang telah dijelaskan tentang UML, bahwa untuk mendapatkan banyak pandangan terhadap sistem informasi yang akan dibangun, UML menyediakan beberapa alat bantu diagram visual yang menunjukan berbagai aspek dalam sistem. Ada beberapa alat bantu diagram yang disediakan dalam

UML antara lain :

1. *Use Case Diagram*

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, kelola data nilai , kelola data absensi, kelola layanan SMS Gateway dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan pekerjaan tertentu.

Use case diagram dapat sangat membantu bila kita sedang menyusun requirement sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang test *case* untuk semua fitur yang ada pada sistem. Sebuah *use case* dapat meng-*include* fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di-*include* akan dipanggil setiap kali *usecase* yang meng-*include* dieksekusi secara normal.

Sebuah *use case* dapat di-*include* oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang *common*. Sebuah *use case* juga dapat meng-*extend use case* lain dengan *behaviour*-nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain.

2. Class Diagram

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). *Class* diagram menggambarkan struktur dan deskripsi class, package dan objek beserta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan lain-lain.

Class memiliki tiga area pokok :

1. Nama (dan stereotype)
2. Atribut
3. Metoda.

Atribut dan metoda dapat memiliki salah satu sifat berikut :

a. *Private,* tidak dapat dipanggil dari luar *class* yang bersangkutan .
b. *Protected,* hanya dapat dipanggil oleh *class* yang bersangkutan dan anak-anak yang mewarisinya.
c. *Public*, dapat dipanggil oleh siapa saja *Class* dapat merupakan implementasi dari sebuah *interface,* yaitu *class* abstrak yang hanya memiliki metoda. *Interface* tidak dapat langsung diinstansiasikan, tetapi harus

diimplementasikan dahulu menjadi sebuah *class*. Dengan demikian *interface* mendukung resolusi metoda pada saat *run-time*. Sesuai dengan perkembangan *class* model, *class* dapat dikelompokkan menjadi *package*. Kita juga dapat membuat diagram yang terdiri atas *package*.

Hubungan Antar *Class* 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 metoda 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. Hubungan dinamis dapat digambarkan dengan menggunakan *sequence* diagram yang akan dijelaskan kemudian.

3. Activity Diagram

Activity diagrams menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, decision yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity* diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity* diagram merupakan state diagram khusus, dimana sebagian besar *state* adalah *action* dan sebagian besar transisi di- *trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu activity diagram tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari *level* atas secara umum.

Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas. Sama seperti *state*, standar UML menggunakan segiempat dengan sudut

membulat untuk menggambarkan aktivitas. *Decision* digunakan untuk menggambarkan *behaviour* pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (*fork* dan *join*) digunakan titik sinkronisasi yang dapat berupa titik, garis *horizontal* atau *vertical*. *Activity* diagram dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu.

Aktifitas diagram ini menggambarkan bagaimana actor seperti guru , siswa, walikelas, administrator berinteraksi dengan sistem dan berintarksi dengan masing-masing aktor agar sesuai dengan kebutuhan yang ada.

4. Sequence Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence* diagram terdiri atar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara *internal* dan *output* apa yang dihasilkan. Masing-masing objek, termasuk aktor, memiliki *lifeline vertikal*. *Message* digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada *fase* desain berikutnya, *message* akan dipetakan menjadi operasi/metoda dari *class*. *Activation bar* menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah *message*.

Alat bantu UML ini akan mempermudah dalam melakukan penelitian untuk lebih memahami apa masalah yang ada, tujuan yang akan dicapai, agar sesuai dengan yang diinginkan pihak klien atau sekolah di SMK Sumatra 40 bandung dan juga mendukung dalam mengembangkan Layanan SMS *Gateway* ini.