

# **RANCANG BANGUN SISTEM INFORMASI PENGOLAHAN DATA ABSENSI SEKOLAH DASAR MENGGUNAKAN BAHASA PEMROGRAMAN JAVA NETBEANS IDE 8.0**

**Dadan Zaliluddin<sup>1</sup>, Shalleha<sup>2</sup>**

*<sup>1</sup>Program Studi Informatika, Fakultas Teknik, Universitas Majalengka  
Jln. KH. Abdul Halim. No. 103 Majalengka 45418  
E-mail : [dadanzuu@gmail.com](mailto:dadanzuu@gmail.com)*

*<sup>2</sup>Program Studi Informatika, Fakultas Teknik, Universitas Majalengka  
Jln. KH. Abdul Halim. No. 103 Majalengka 45418  
E-mail : [shalleha1995@gmail.com](mailto:shalleha1995@gmail.com)*

## **ABSTRAK**

Absensi merupakan hal yang wajib dilakukan untuk siswa. Dalam melakukan absensi masih menggunakan cara manual. Oleh karena itu, diperlukan suatu sistem absensi siswa yang dapat melakukan pendataan dan pengelolaan data agar dapat dilakukan secara cepat, efisien, dan akurat. Pembuatan sistem absensi siswa ini dilakukan dengan cara pengumpulan data, analisa sistem, perancangan sistem, dan implementasi sistem. Sistem ini dibuat dengan menggunakan bahasa pemrograman Java Netbeans IDE 8.0 dan MySQL untuk pengelolaan database. Sebagai hasil dari tugas akhir ini adalah dibuatnya aplikasi berbasis dekstop Sistem Informasi Absensi siswa Berbasis dekstop Menggunakan Java Netbeans IDE 8.0 yang memiliki fasilitas seperti: manajemen data siswa, manajemen data guru, manajemen absensi siswa, manajemen absensi guru. Dengan adanya aplikasi ini, diharapkan proses absensi akan lebih efisien dan dapat dipantau dengan mudah oleh dosen dan oleh pusat.

### **Kata Kunci**

Absensi, Aplikasi, Java Netbeans, Sistem Informasi.

## 1. PENDAHULUAN

Perkembangan ilmu dan teknologi yang sangat maju ini berpengaruh terhadap kemajuan dibidang pendidikan. Untuk memajukan pendidikan diperlukan dukungan manajemen yang tepat dan dalam mengelolanya, diperlukan informasi yang teliti, tepat dan cepat. Demikianlah kecenderungan pendidikan dizaman modern ini, berkembang pesat dan penuh persaingan serta maju mundurnya sangat tergantung kepada informasi yang ditangani dengan baik. SDN Pelasahkoneng adalah salah satu sekolah dasar yang terletak di Kecamatan Gantar Kabupaten Indramayu Provinsi Jawa Barat, sistem absensi yang ada disekolah tersebut masih menggunakan sistem manual sehingga membutuhkan suatu terobosan baru dalam proses absensi, maka aplikasi sistem informasi pengolahan data absensi sekolah dasar negeri Pelasahkoneng menggunakan bahasa pemrograman *java netbeans ide 8.0* yang bertujuan untuk mengatasi masalah yang ada di SDN Pelasahkoneng ini, untuk mengubah penggunaan absensi secara manual menjadi lebih efektif dan efisien. Hal ini mendorong penulis untuk mencoba membuat aplikasi sistem informasi pengolahan data absensi yang dapat membantu petugas dalam hal absensi, sehingga akan meningkatkan kinerja petugas.

## 2. METODE PENELITIAN

Model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linier*) atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (Rosa & Shalahuddin, 2013).

Tahap atau fase yang terdapat dalam Waterfall adalah sebagai berikut [3]:

1. Analisis kebutuhan perangkat lunak  
Proses pengumpulan kebutuhan dilakukan secara intensip untuk mespesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.
2. Desain  
Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengidean. Tahap ini mentranslasi kebutuhan

perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

3. Pembuatan kode program  
Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.
4. Pengujian  
Pengujian fokus pada perangkat lunak secara dari segi logik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.
5. Pendukung (*support*) atau Pemeliharaan (*maintenance*)  
Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru [6].

### 2.1 Pengenalan Unified Modeling Language (UML)

Pada perkembangan teknik pemrograman berorientasi objek, muncullah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun menggunakan teknik pemrograman berorientasi objek, yaitu *Unified Modeling Language (UML)*. UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung [7].

#### 1. Use Case Diagram

*Use Case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang

akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut.

### 2. Class Diagram

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Diagram kelas dibuat agar pembuatan program atau *programmer* membuat kelas-kelas sesuai rancangan di dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron.

### 3. Activity Diagram

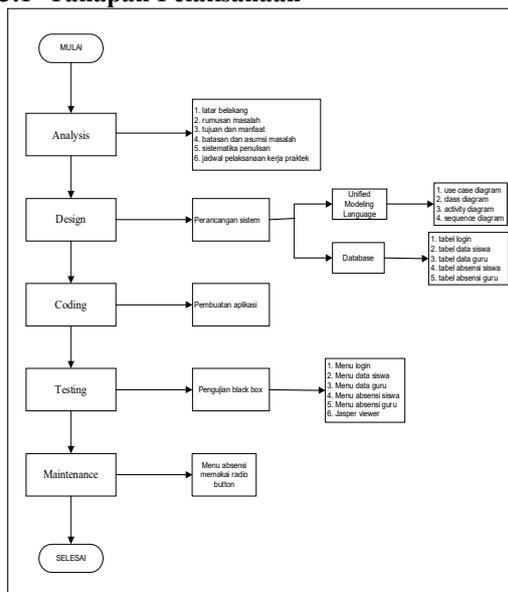
Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau *menu* yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan sistem.

### 4. Sequence Diagram

Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram sekuen maka harus diketahui objek-objek yang terlihat dalam *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*.

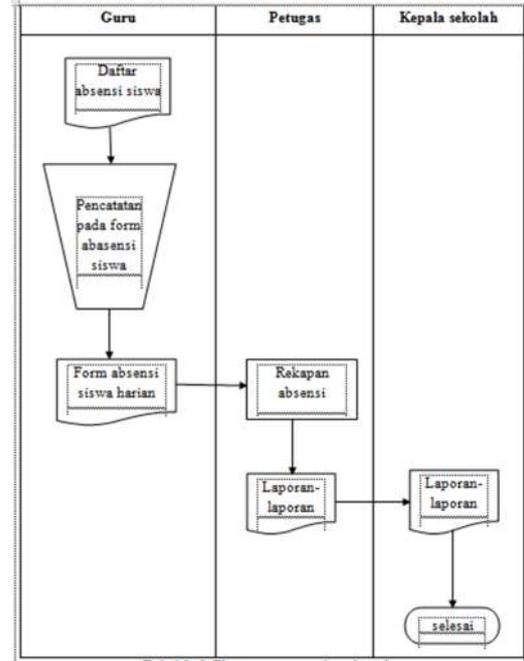
## 3. HASIL DAN PEMBAHASAN

### 3.1 Tahapan Pelaksanaan



Gambar 3.1 Tahapan Pelaksanaan

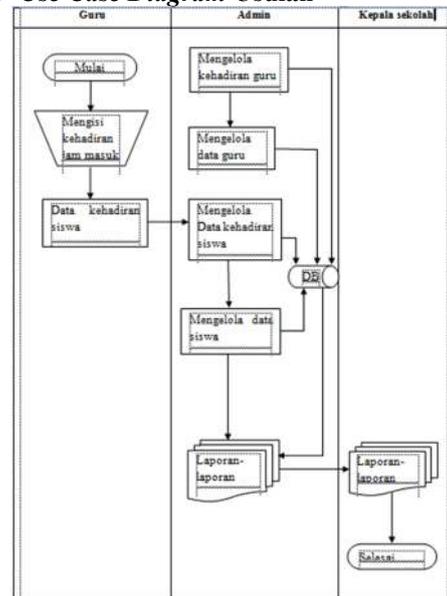
### 3.2 Analisis Sistem yang Sedang Berjalan



Gambar 3.2 Alur kegiatan yang sedang berjalan

Pada gambar 3.2 menjelaskan tentang alur sistem yang sedang berjalan dimana dalam pengolahan absensi masih dilakukan secara manual dan disimpan dalam bentuk arsip.

### 3.3 Use Case Diagram Usulan

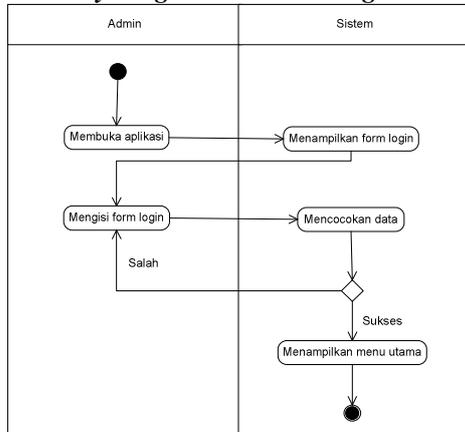


Gambar 3.3 Alur yang sedang diusulkan

Pada gambar 3.3 menjelaskan tentang alur sistem yang diusulkan dimana dalam pengolahan data kependudukan sudah dilakukan secara terkomputerisasi dilakukan dengan menggunakan aplikasi dan datanya

langsung disimpan pada penyimpanan database.

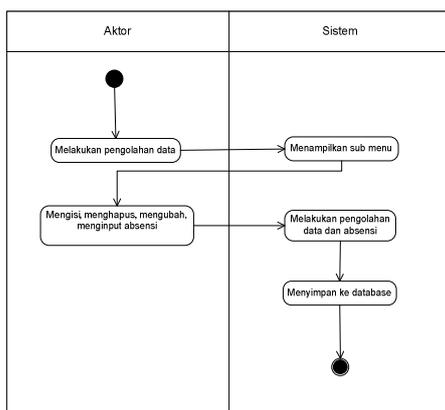
### 3.4 Activity Diagram Halaman Login



Gambar 3. 4 Activity Diagram Login

Gambar diatas merupakan aktivitas *login* admin yang menggambarkan proses *login* admin. Dimulai dengan admin membuka aplikasi data kategori kesejahteraan penduduk kemudian sistem menampilkan *form login*, selanjutnya admin dapat melakukan *login* dengan memasukan *username* dan *password* untuk dapat masuk kedalam sistem. Proses login itu sendiri diikuti juga validasi yang dilakukan oleh sistem dimana terdapat proses verifikasi *username* dan *password* apakah sesuai atau tidak. Jika sesuai admin tersebut dapat masuk menggunakan sistem atau jika tidak admin akan diminta memasukan *username* dan *password* kembali.

### 3.5 Activity Diagram Pengolahan Data

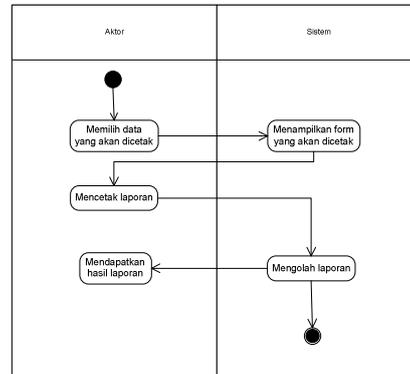


Gambar 3. 5 Activity Diagram Login

Gambar diatas menunjukan aktivitas pengelolaan data absensi sekolah dasar pelaskoneng yang dilakukan oleh admin. Dimulai dari admin melakukan pengolahan data kemudian sistem menampilkan *sub menu*,

selanjutnya *admin* dapat mengisi, mengubah dan menghapus data lalu sistem melanjutkan pengolahan data absensi sekolah dasar pelaskoneng yaitu tersimpannya data kedalam *database*.

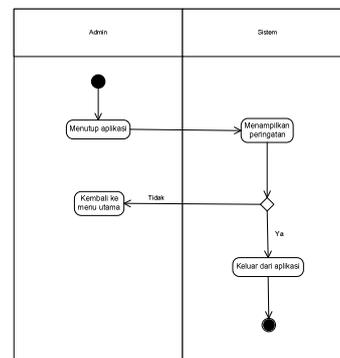
### 3.6 Activity Diagram Laporan



Gambar 3. 6 Activity Diagram Laporan

Gambar diatas menunjukan aktivitas laporan yang dilakukan oleh *admin*. Dimulai dari admin memilih data absensi siswa dan guru kemudian sistem menampilkan data absensi siswa dan guru yang dipilih, selanjutnya *admin* mencetak laporan lalu sistem mengolah laporan dan admin mendapatkan hasil cetakan laporan.

### 3.7 Activity Diagram Logout

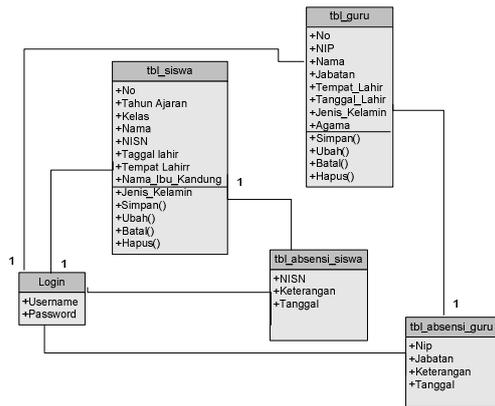


Gambar 3. 7 Activity Diagram Logout

Gambar diatas menunjukan aktivitas *logout* yang dilakukan oleh admin. Dimulai dari *admin* menutup aplikasi kemudian sistem menampilkan peringatan berupa *message box*. Jika tidak akan kembali ke *menu* utama atau jika ya admin langsung keluar dari aplikasi.

### 3.8 Class Diagram

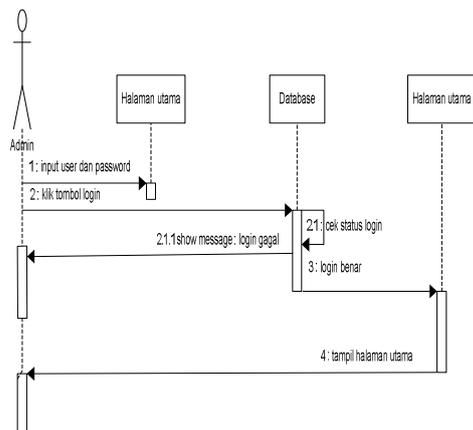
Diagram kelas menggambarkan kelas-kelas yang saling berinteraksi. *Admin* berinteraksi dengan data siswa, data guru, dan absensi. Terlihat pada Gambar di bawah ini.



Gambar 3.8 Class Diagram

### 3.9 Sequence Diagram Halaman Login

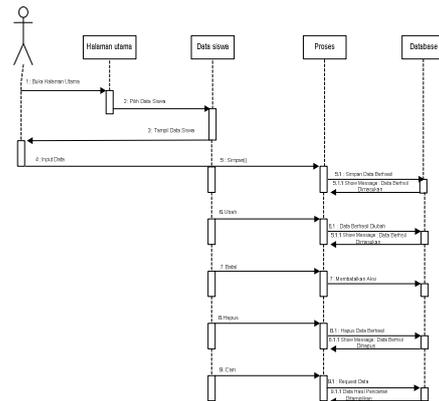
Sequence diagram Login merupakan proses dimana admin melakukan login terlebih dahulu sebelum masuk ke halaman utama untuk mengelola data. Sequence diagram login terlihat pada Gambar dibawah ini.



Gambar 3.9 Sequence Diagram Login

### 3.10 Sequence Diagram Data Siswa

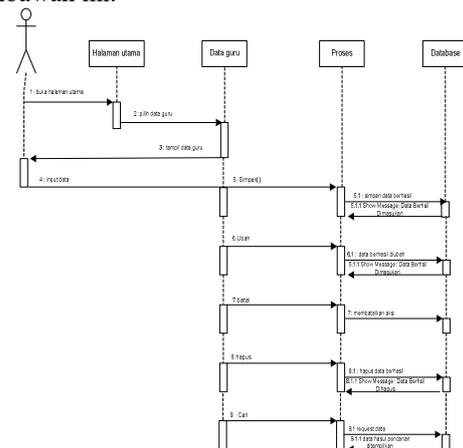
Sequence diagram dari detail data siswa ini menjelaskan proses dalam mengelola data siswa. Admin dapat menambahkan data siswa baru ke dalam database dengan mengisi form data siswa yang disediakan, Untuk mengubah dan menghapus data siswa yang sudah tersimpan di database. Seperti terlihat pada Gambar 4.8. dibawah ini.



Gambar 3.10 Sequence Data Siswa

### 3.11 Sequence Diagram Data Guru

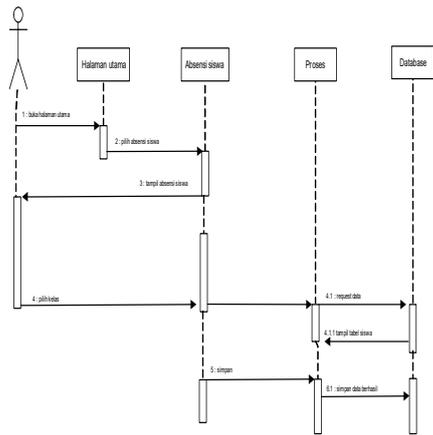
Sequence diagram dari detail data guru ini menjelaskan proses dalam mengelola data guru. Admin dapat menambahkan data guru ke dalam database dengan mengisi form data guru yang disediakan, untuk mengubah dan menghapus data guru yang sudah tersimpan di database. Seperti terlihat pada Gambar 4.9. dibawah ini.



Gambar 3.11 Sequence Diagram Data Guru

### 3.15 Sequence Diagram Absensi Siswa

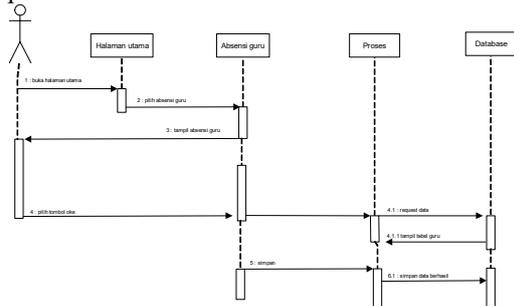
Sequence diagram dari absensi siswa ini menjelaskan proses dalam mengelola absensi siswa. Admin dapat mengabsen siswa secara otomatis sesuai dengan kelasnya masing-masing, dan mencetak data absensi siswa yang sudah tersimpan di database. Seperti terlihat pada Gambar 4.10. dibawah ini.



Gambar 3.12 sequence diagram absensi siswa

### 3.16 Sequence Diagram Abseni Guru

Sequence diagram dari absensi guru ini menjelaskan proses dalam mengelola absensi guru. Admin dapat mengabsen guru secara otomatis, dan mencetak data absensi guru yang sudah tersimpan di database. Seperti terlihat pada Gambar 4.11. dibawah ini.



Gambar 3.13 sequence diagram absensi guru

### 3.1 Tampilan Halaman Login



Gambar 4.1 Tampilan Halaman Login

### 3.2 Tampilan Menu Utama



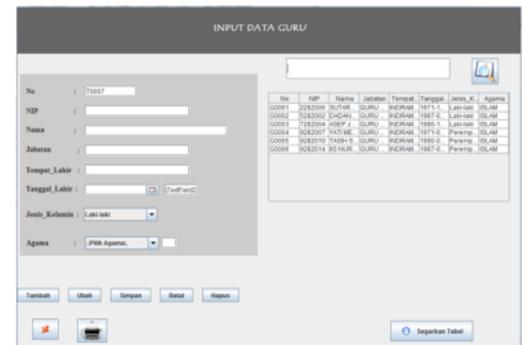
Gambar 4.2 Tampilan Menu Utama

### 3. Tampilan Menu Data siswa



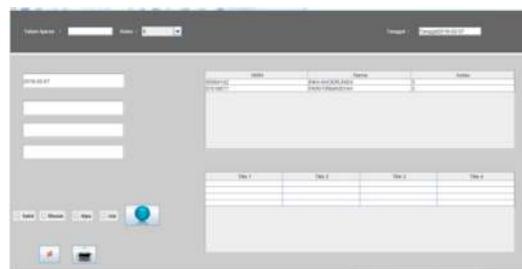
Gambar 4.3 Tampilan Menu Data Siswa

### 4. Tampilan Menu Data Guru



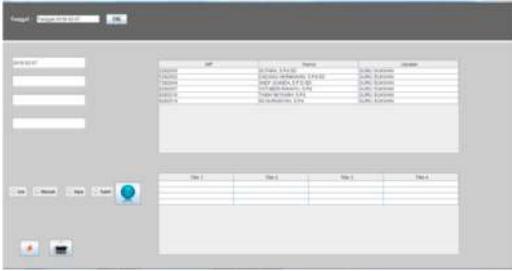
Gambar 4.4 Tampilan Menu Data Guru

### 5. Tampilan Menu Absensi Siswa



Gambar 4.5 Tampilan Menu Absensi Siswa

## 6. Tampilan Menu Absensi Guru



Gambar 4.6 Tampilan Menu Absensi Guru

## 7. Tampilan Menu Laporan

NIS	NAMA	KELAS	KETERANGAN	TANGGAL
87470260	ALYA FATMAWATI	4	Apa	9/10/19 12:00 AM
8222092	KARWANAN	2	Apa	9/10/19 12:00 AM
88807129	TOMI KURNIAWAN	3	Sakit	9/10/19 12:00 AM
6000294	ANDI RUSMAN	5	Masuk	9/10/19 12:00 AM
80031793	DHELVA SARYA	2	Apa	9/10/19 12:00 AM
98970993	SRI ELANI	2	Masuk	9/10/19 12:00 AM
5108863	CAHYA ROSE	6	Apa	9/10/19 12:00 AM
8307007	WULAN FITRIANATI	3	Masuk	9/10/19 12:00 AM
88801819	KADAH AULIA	2	Masuk	9/10/19 12:00 AM
88731076	REKI FIRMANDEWAN	2	Sakit	9/10/19 12:00 AM
72027000	AYU	4	Masuk	9/10/19 12:00 AM
82470007	DEWANGDA	3	Sakit	9/10/19 12:00 AM
82080790	NANDANG HENDI	2	Sakit	9/10/19 12:00 AM
82540008	WAFU SURYA	3	Masuk	9/10/19 12:00 AM

Gambar 4.7 Tampilan Menu Laporan

## 4. KESIMPULAN

### Kesimpulan

1. Pembuatan aplikasi absensi sekolah menggunakan java netbeans yaitu pertama kita harus menginstal aplikasi netbeans 8.0, untuk penyimpanan database gunakan aplikasi XAMPP, sebaiknya sebelum memulai pada pembuatan aplikasi akan lebih baik membuat database terlebih dahulu, pembuatan desain aplikasi, dan membuat coding pada aplikasi yang akan dibuat.
2. Penerapan sistem absensi manual kedalam sistem aplikasi dengan membuat database absensi agar dalam penerapan ke dalam aplikasi lebih mudah karena setiap data sudah bisa diinput di aplikasinya begitupun dengan laporan.

3. Penyimpanan data disimpan didalam database menggunakan phpMyadmin.

### Saran

Penelitian ini masih terdapat kekurangan, adapun saran untuk pengembangan aplikasi selanjutnya :

Dengan adanya sistem informasi aplikasi absensi ini, diharapkan ada upaya pengembangan lebih lanjut menjadi aplikasi sistem informasi aplikasi absensi yang lebih sempurna dengan fitur-fitur terbaru yang nantinya akan semakin memudahkan dan belum dibuatkannya laporan absensi untuk data perbulan sehingga diharapkan untuk dapat mengembangkan penelitian selanjutnya.

## 5. REFERENSI

- [1] sari, i. p. (t.thn.). Dipetik juni 17, 2013, dari pengertian rancang bangun dan konsep sistem informasi.
- [2] Fathansyah. (2012). *Basis Data*. Bandung: Informatika Bandung.
- [3] Indrajadi. (2014). *Database system case study all in one*. Jakarta: PT Elex Media Komputido.
- [4] Jogyanto. (2005). *Analisis dan Desain Sistem Informasi*. Yogyakarta: Penerbit Andi.
- [5] Ladjamudin, A.-B. (2005). *Analisis dan Desain Sistem Informasi*. Yogyakarta: Graha Ilmu
- [6] Rosa, A. S., & Shalahuddin, M. (2013). *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: Informatika Bandung.
- [7] W, P. P., & Herlawati. (2011). *Menggunakan UML*. Oktober: Informatika Bandung.