BAB III

OBJEK DAN METODE PENELITIAN

3.1 Objek penelitian

Pada penulisan tugas akhir ini penulis melakukan penelitian di SDN PABAKI 3 kota Bandung. Adapun yang akan dibahas pada objek penelitian ini yaitu mengenai sejarah singkat SDN PABAKI 3, visi dan misi SDN PABAKI 3 , struktur organisasi dan deskripsi tugas yang terdapat pada SDN PABAKI 3.

3.1.1 Sejarah Singkat SDN Pabaki 3

SDN Pabaki 3 berdiri pada tahun 1920 dengan wilayah tanah seluas 2207 m^2. Dari luas tanah tersebut dibangun beberapa ruang diantaranya :

1. Ruangan Kantor Kepala Sekolah
2. Ruang kerja kelompok guru
3. Ruang UKS
4. Ruang pramuka
5. Ruang perpustakaan
6. Ruang kelas yang berjumlah 20
7. Mesjid
8. Lapangan

Mulai dari tahun 1920 sampai 2012 banyak sekali perubahan seperti mutu pendidikan dan perbaikan gedung – gedung sekolah yang baru.

3.1.2 Visi dan Misi Sekolah

Visi dan Misi di SDN PABAKI 3 adalah :

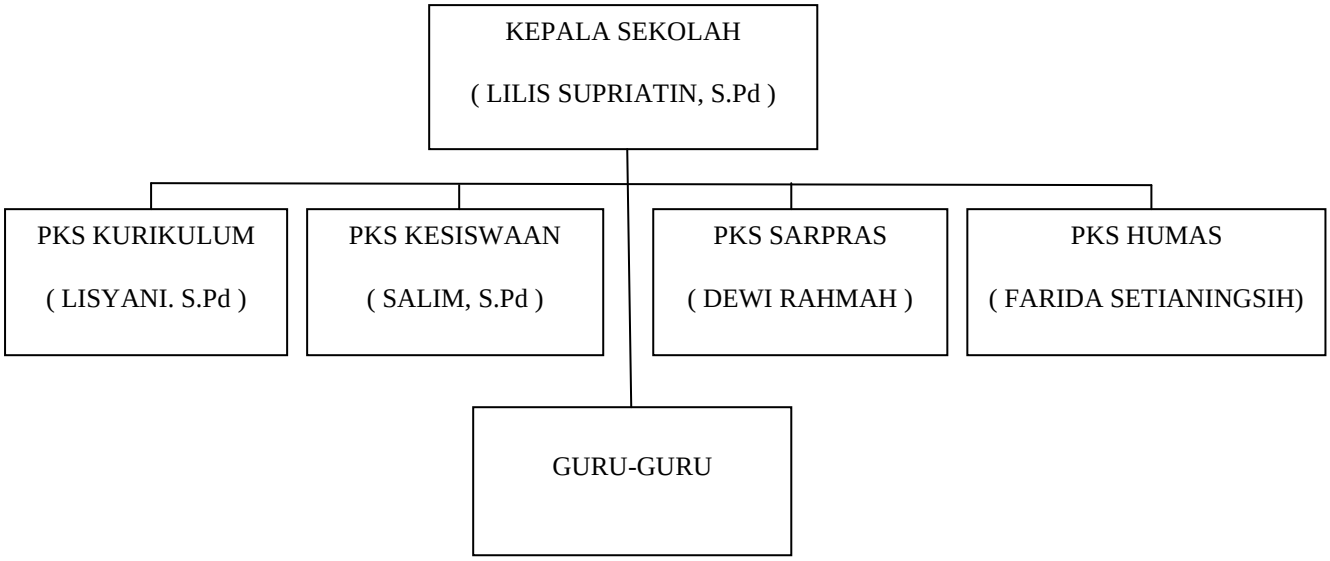
1. Visi

Menciptakan lulusan sekolah dasar negeri Pabaki 3 yang berbudi, berprestasi dan berteknologi.

2. Misi

   a. Mengembangan perangkat pembelajaran yang benuansa agama, mengembangkan nilai-nilai moral dan bermuatan teknologi.
   b. Mengembangkan perangkat pembelajaran yang berbasis pembelajaran yang aktif, kreatif, efektif, inovatif, dan menyenangkan.
   c. Mengembangkan kegiatan keagamaan untuk pembinaan pribadi muslim yang baik.
   d. Mengembangkan bakat, minat, dan potensi peserta secara maksimal melalui kegiatan ekstrakulikuler.
   e. Mengembangkan dan membiasakan perilaku disiplin warga.

3.1.3 Struktur Organisasi Sekolah



Gambar 3.1 Struktur Organisasi SDN PABAKI 3
Sumber : (Dokumen SDN PABAKI 3).

3.1.4 Deskripsi Tugas

1. Kepala Sekolah bertugas membimbing dan memimpin guru dalam proses kegiatan belajar mengajar.
2. PKS Kurikulum bertugas melaksanakan administrasi dan mengelola kurikulum.
3. PKS Kesiswaan bertugas melaksanakan administrasi dan mengelola kegiatan kesiswaan.
4. PKS Sarana dan prasarana bertugas melaksanakan administrasi dan mengelola sarana prasarana.

5. PKS Hubungan Masyarakat bertugas melaksanakan administrasi dan fungsi humas.

3.2 Metode Penelitian

Dalam penelitian ini, peneliti menggunakan rancangan penelitian berdasarkan metode deskriptif dan action.

Metodologi penelitian yang digunakan adalah metode **deskriptif** dengan jenis penelitian studi kasus. Metode deskriptif adalah suatu metoda dalam meneliti status sekelompok manusia, suatu objek, suatu set kondisi, suatu sistem pemikiran ataupun suatu peristiwa pada masa sekarang.

3.2.1 Desain Penelitian

Dalam penelitian ini, peneliti menggunakan rancangan penelitian berdasarkan metode **deskriptif** yaitu suatu metode dalam meneliti status kelompok manusia, suatu obyek, suatu set kondisi, suatu sistem pemikiran ataupun suatu kelas peristiwa pada masa sekarang.

3.2.2 Jenis dan Metode Pengumpulan Data

Jenis dan metode pengumpulan data terdiri dari sumber data primer dan sumber data sekunder.

3.2.2.1 Sumber Data Primer (Wawancara, Observasi, Kuesioner)

Sumber data primer adalah data yang didapat langsung dari sumber nya (*responden). Responden* adalah pemberi informasi yang diharapkan dapat

menjawab semua pertanyaan dengan jelas dan lengkap. Dalam hal ini penulis memperoleh sumber data dari *responden* yang terkait dengan objek penelitian, yaitu *Guru* dalam hal ini yaitu guru dan *Siswa* SDN PABAKI 3 . Adapun metode pengumpulan data dalam penelitian ini yaitu sebagai berikut :

1. **Wawancara (*interview*)**

Merupakan teknik pengumpulan data untuk mendapatkan informasi langsung dari sumbernya secara tatap muka dan wawancara langsung dengan objek penelitian (*responden*). Dalam hal ini penulis melakukan wawancara dengan yaitu *Kepala Sekolah* dan *Guru* SDN PABAKI 3 langsung. Wawancara ini dilakukan untuk mendapatkan informasi tentang seberapa jauh siswa mengetahui tata cara shalat pada khususnya dan pendidikan agama pada umumnya.

2. **Observasi**

Observasi adalah suatu pengamatan sistematis yang dilakukan secara langsung terhadap objek di tempat penelitian. Dalam hal ini penulis melakukan observasi dengan mengamati kegiatan belajar mengajar di SDN PABAKI 3.

3.2.2.2 Sumber Data Sekunder

Adapun data yang berasal dari sumber data sekunder diperoleh dengan teknik dokumentasi. Dokumentasi adalah teknik pengumpulan data dengan cara mengumpulkan dokumen-dokumen yang berhubungan dengan objek penelitian.

Dalam hal ini, dokumen-dokumen yang diperoleh dianalisis sehingga diperoleh data-data yang sesuai untuk kegiatan pengembangan sistem.

3.2.3 Metode Pendekatan dan Pengembangan Sistem

3.2.3.1 Metode Pendekatan

Metode pendekatan sistem adalah metode yang memandang software atau aplikasi sebagai sebuah interaksi antar bagian dalam sebuah sistem. Dalam hal ini metode pendekatan berorientasi objek.

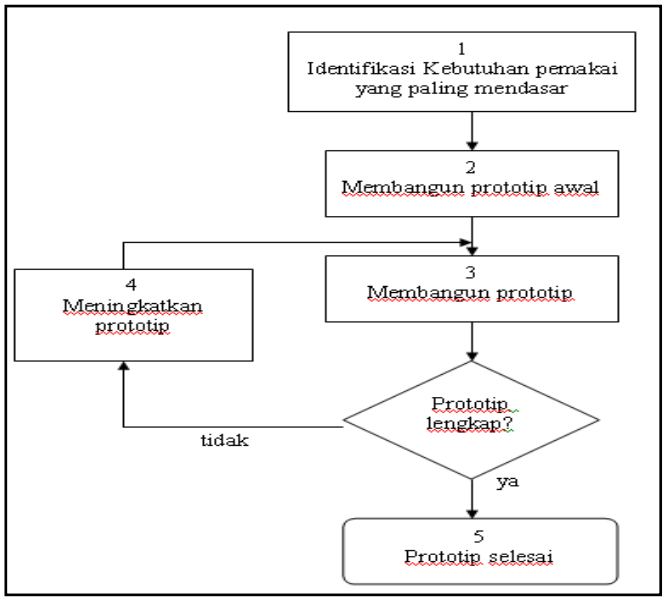
3.2.3.2 Metode Pengembangan

Metode pengembangan sistem adalah metode-metode, prosedur-prosedur, konsep-konsep pekerjaan dan aturan-aturan untuk mengembangkan suatu sistem informasi. Dengan mengikuti metode dan prosedur-prosedur yang diberikan oleh suatu metode, maka pengembangan sistem diharapkan akan dapat diselesaikan dengan berhasil.

Metode penelitian khusus untuk perangkat lunak adalah metode *protoype*, *prototyping* merupakan suatu model yang memperlihatkan fitur-fitur suatu produk, layanan, atau sistem usulan. Modelnya dikenal dengan sebutan metode *prototype*. Metode penelitian ini dipilih karena alasan sebagai berikut :

Protoype memberikan ide bagi disainer sistem maupun *user* tentang cara sistem akan berfungsi dalam bentuk lengkapnya. Langkah dalam *prototyping* adalah seperti berikut :

1. Mengidentifikasi kebutuhan-kebutuhan dasar pemakai. Dalam hal ini perancangan sistem bekerja dengan pemakai untuk menangkap informasi dasar yang diperlukan pemakai.
2. Mengembangkan sebuah *prototype*. Perancangan sistem menciptakan sebuah *prototype* dengan cepat. Dengan menggunakan perangkat lunak generasi keempat atau menggunakan perangkat lain. *Prototype* dapat hanya mancakup fungsi-fungsi yang paling penting atau mencakup seluruh sistem.
3. Menggunakan *prototype*. Pada tahapan ini, pemakai diminta untuk bekerja dengan sistem untuk menentukan cocok-tidaknya *prototype* terhadap kebutuhan pemakai dan diharapkan pemakai memberi saran-saran untuk perbaikan *prototype*.
4. Memperbaiki dan meningkatan *Prototype*. *Prototype* diperbaiki sesuai dengan semua perubahan yang diminta atau yang disarankan oleh pemakai. Setelah diperbaiki, langkah 3 dan 4 dilakukan secara terus menerus sampai pemakai merasa puas.



Gambar 3.2 Membangun *Prototype*
(Sumber: Jogiyanto HM,2005, Sistem Teknologi Informasi,Andi)

3.2.3.3 Alat Bantu Analisis Dan Perancangan

Alat bantu yang digunakan dalama analisis dan perancangan sistem ini adalah sebagai berikut:

1. Use Case

Use Case adalah deskripsi fungsi dari sebuah sistem dari perspektif pengguna. Use Case bekerja dengan cara mendeskripsikan tipikal interaksi antara user (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Urutan langkah-langkah yang menerangkan antara pengguna dan sistem disebut *scenario*. Setiap *scenario* mendeskripsikan urutan kejadian. Setiap urutan diinisialisasi oleh orang, sistem yang lain, perangkat keras atau urutan waktu.

Dengan demikian secara singkat bisa dikatakan use case adalah serangkaian *scenario* yang digabungkan bersama-sama oleh tujuan umum pengguna. Dalam pembicaraan tentang *use case*, pengguna biasanya disebut dengan aktor. Aktor adalah sebuah peran yang bisa dimainkan oleh pengguna dalam interaksinya dengan sistem.

Model *use case* adalah bagian dari model *requirement* (Jacob et all, 1992). Termasuk disini adalah *problem* domain *object* model dan penjelasan tentang user *interface*. *Use case* memberikan spesifikasi fungsi-fungsi yang ditawarkan oleh sistem dari perspektif *user*.

2. Sequence Diagram

Sequence diagram digunakan untuk menggambarkan perilaku pada sebuah skenario. Diagram ini menunjukkan sejumlah contoh obyek dan *message* yang

diletakkan diantara obyek-obyek ini di dalam use case. Sequence diagram menambahkan dimensi waktu pada interaksi diantara obyek. Pada diagram ini *participant* diletakkan di atas dan waktu ditunjukkan dari atas ke bawah. *Lifeline participant* diurutkan dari setiap participant. Kotak kecil pada *Lifeline* menyatakan activation, yaitu menjalankan salah satu operation dari participant. State bias ditambahkan dengan menambahkannya sepanjang life line.

Message (sederhana, *synchronous* atau *asynchronous*) adalah tanda panah yang menghubungkan suatu *life line* ke *life line* yang lain. Lokasi *life line* dalam dimensi vertikal mewakili urutan waktu dalam *sequence* diagram. *Message* yang pertama terjadi adalah yang paling dekat dengan bagian atas diagram dan yang terjadi belakangan adalah yang dekat dengan bagian bawah.

3. Activity Diagram

Activity diagram seperti sebuah *flowchart*. *Activity* diagram menunjukkan tahapan, pengambilan keputusan dan percabangan. Diagram ini sangat berguna untuk menunjukkan operation sebuah obyek dan proses bisnis. Kelebihan *activity* diagram dibandingkan *flowchart* adalah kemampuannya dalam menampilkan aktivitas parallel.

4. Component Diagram

Component diagram merepresentasikan dunia rill item yaitu component *software*. Component software menetap di komputer bukan di benak para analis.

Component bisa diakses melalui *interface* nya yaitu koleksi operasi-operasi. Relasi antara *component* dan interfacenya disebut *realization*. Suatu

component bias mengakses *service-service* yang ada di *component* lain dengan cara *import interface*. Sedangkan *component* yang menyediakan *service* menggunakan *export interface*.

5. Deployment Diagram

Deployement diagram menyediakan gambaran bagaimana sistem secara fisik akan terlihat. Sistem terdiri dari node-node dimana setiap node diwakili untuk sebuah kubus. Garis yang menghubungkan antara 2 kubus menunjukkan hubungan diantara kedua node tersebut. Tipe node bisa berupa *device* yang berwujud *hardware* dan bisa juga *processor* (yang mengeksekusi *component*) atau *execution environment (software* yang menjadi *host* atau mengandung *software* yang lain).

6. Class Diagram

Kotak adalah notasi UML untuk *class*. Nama, *attribute, operation* dan *responsibility* dari *class* ada pada kotak tersebut. *Stereotype* bisa dipergunakan untuk mengorganisasikan daftar *attribute* dan *operation*. Dalam beberapa kasus, kadangkala hanya perlu ditampilkan sebagian saja dari *attribute* dan *operation*. Tipe *attribute* dan nilai *default* bisa dimunculkan sebagaimana pada *operation*. Untuk mengurangi ambiguitas pada pendeskripsian *class*, *constraint* bisa ditambahkan. Bahkan kalau perlu bisa ditambahkan *attached notes* ke dalam kotak tersebut.

3.2.4 Pengujian Software

Pada pengujian *black-box*, kasus-kasus pengujian berdasarkan pada spesifikasi sistem. Rencana pengujian dapat dimulai sedini mungkin di proses pengembangan perangkat lunak.

Pada pengujian *black box*, mencoba beragam masukan dan memeriksa keluaran yang dihasilkan. Teknik pengujian *black box* juga dapat digunakan untuk pengujian berbasis skenario, dimana isi dalam sistem mungkin tidak tersedia untuk diinspeksi tapi masukan dan keluaran yang didefinisikan dengan *use case* dan informasi analisis yang lain.

Adapun faktor-faktor pengujian *black-box* adalah :

1. ***File integrity***

Menekankan pada data yang dimasukkan melalui aplikasi akan tidak bias diubah. Prosedur yang akan memastikan bahwa *file* yang digunakan benar dan data dalam *file* tersebut akan disimpan sekuensial dan benar.

2. ***Service levels***

Menekankan bahwa hasil yang diinginkan didapat dalam waktu yang diinginkan oleh *user*. Untuk mencapai keinginan tersebut, harus dilakukan penyesuaian antara keinginan *user* dengan sumber daya yang ada.

3. ***Ease of use***

Menekankan perluasan usaha yang diminta untuk belajar, mengoprasikan dan menyiapakan inputan, dan menginterpretasikan output dari sistem. Faktor ini tersangkut dengan *usability* system terhadap interaksi antara manusia dan system.

4. *Authorization*

Menjamin data diproses sesuai dengan ketentuan manajemen. *Authorization* menyangkut proses transaksi secara umum dan khusus.

Fokus Pengujian *Black box testing* yaitu sebagai berikut :

1. Menguji fungsi-fungsi khusus dari aplikasi.
2. *Test input* dan *output* untuk fungsi yang ada tanpa memperhatikan prosesnya.

Beberapa jenis kesalahan yang dapat di identifikasi :

1. Fungsi tidak benar atau hilang,
2. Kesalahan antar muka,
3. Kesalahan pada struktur data (pengaksesan basis data),
4. Kesalahan inisialisasi dan akhir program, dan
5. Kesalahan performasi.