BAB III

METODOLOGI PENELITIAN

3.1.Objek Penelitian

3.1.1. Sejarah Perusahaan

RA Bina Qur'ani Nur'aini didirikan tahun 2000 diatas tanah wakaf dari salah seorang warga bojong koneng dengan ukuran ± 7×3 m² yang peduli pada pendidikan di wilayah bojong koneng . Sebagai ketua yayasan adalah Ketua DKM mesjid Nur'aini pada waktu itu beserta pengurus dan jajarannya mendirikan Yayasan Pendidikan yang dikelola oleh bapak Engkus sebagai kepala RA, namun kesadaran masyarakat pada saat itu akan pentingnya pendidikan anak usia dini masih kurang. Barulah pada tahun 2005 kesadaran masyarakat dilingkungan RA Bina Qur'ani Nur'aini sudah mulai meningkat, hal ini terlihat dari meningkatnya jumlah peserta didik dari tahun 2005 hingga sekarang.

Alasan Kami mendirikan RA, kami merasa tergugah dan terpanggil untuk memajukan RA yang ada di lingkungan masyarakat Bojong Koneng .

Dengan sarana dan prasarana seadanya, mengapa mereka tidak masuk RA dulu sebelum ke tingkat SD ? namun menurut masyarakat di Bojong Koneng saat itu biaya untuk masuk RA atau TK tidak terjangkau oleh mereka, sehingga mereka memilih untuk langsung

memasukkan para putra putri mereka ke SD yang jaraknya tidak ada yang dekat.

Setelah kami mengadakan pendekatan dengan jajaran pengurus daerah (Rt, Rw , Pengurus PosYandu) yang ada tersebut, mereka sangat mendukung . dengan hanya satu ruang kelas kami dapat mengatur pembelajaran menjadi dua kelompok, yaitu : kelompok A masuk pagi dan Kelompok B masuk siang. Dengan SPP yang cukup terjangkau masyarakat, Alhamdulillah murid kami tidak berkurang sampai sekarang dan setiap tahun meningkat terus sampai tidak tertampung, walaupun dilingkukngan kami dekat dengan RA dan TK lain.

Harapan kami mendirikan RA ini, dapat meningkatnya mutu pendidikan, baik dari segi agama, akhlaq, adzab sopan santun, motorik, kognitif dan bahasa anak – anak generasi penerus khususnya dilingkungan Bojong Koneng.

3.1.2. Visi Misi Perusahaan

3.1.2.1. Visi

Cerdas,Terampi,Mandiri,Berakhlaq Mulia.

3.1.2.2. Misi

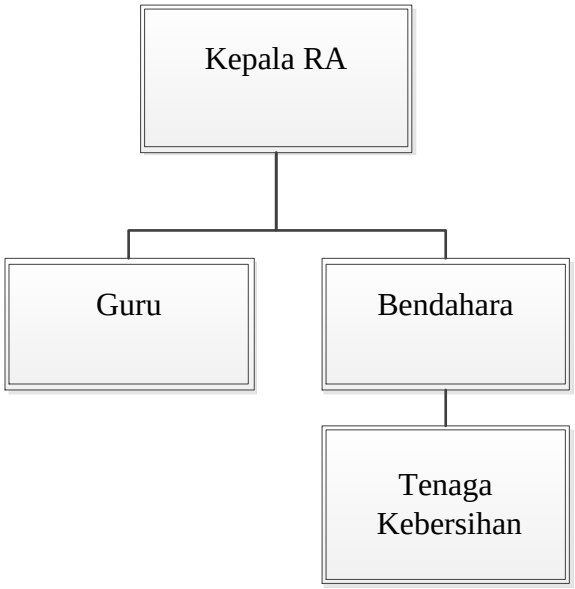
Mengembangkan sikap dan pengetahuan yang luas.

Mengembangkan keterampilan dan daya cipta sedini mungkin.

Mengarahkan peserta didik agar dapat menyesuaikan diri dengan lingkungannya.

Mengembangkan potensi peserta didik agar menjadi anak sholeh, beriman, bertaqwa dan berakhlaqul mulia.

3.1.3. Struktur Organisasi



Gambar 3.1 Struktur Organisasi Umbara Trans

3.1.4. Deskrpsi Kerja

a. Kepala RA : Sebagai penggung jawab setiap kegiatan belajar mengajar di raudhatul athfal bina qur'ani nur'aini.
b. Guru : Sebagai tenaga pengajar di raudhatul athfal bina qur'ani nur'aini.

c. Bendahara : Sebagai yang mengatur setiap kegiatan oprasional di raudhatul athfal bina qur'ani nur'aini.
d. Tenaga Kebersihan : Bertangung jawab dengan kebersihan di raudhatul athfal bina qur'ani nur'aini.

3.2.Metode Penelitian

3.2.1. Desain Penelitian

Desain penelitian yang digunakan dalam penelitian ini adalah deskriptif dan action. Selain penelitian deskriptif juga menggunakan desain penelitian action.

Penelitian deskriptif dipakai penulis untuk menggambarkan dan memaparkan keadaan dari sistem saat ini atau menggambarkan yang terjadi pada sistem khususnya adalah menggambarkan aplikasi edukasi bagi anak usia dini berbasis android.

Penelitian Action dipakai peneliti untuk bertindak memecahkan masalah yang terjadi pada sistem pembelajaran pada anak usia dini yang ada saat ini dengan memberikan solusi atau ide baru untuk memperbaikinya.

3.2.2. Jenis dan Metode Pengumpulan Data

3.2.2.1. Sumber Data Primer

Sumber data primer didapat dari beberapa cara antara lain adalah :

a. Observasi

Metode pengumpulan data yang dilakukan dengan mengamati secara langsung anak – anak usia dini dan guru – guru melakukan proses belajar mengajar

b. Wawancara

Metode pengumpulan data yang dilakukan dengan bertanya langsung ke sumber sumber ahli dalam bidang edukasi pada anak usia dini sperti guru –guru tk atau paud.

3.2.2.2. Sumber Data Sekunder

Data sekunder dalam penelitian ini berdasarkan data – data yang berhasil dikumpulkan dari internet dan buku – buku yang membahas tentang pemograman android.

3.2.3. Metode Pendekatan dan Pengembangan Sistem

3.2.3.1. Metode Pendekatan

Metode pendekatan sistem yang dipakai adalah *Object Oriented Development*, yaitu suatu cara pengembangan perangkat lunak dan sistem informasi berdasarkan abstraksi objek-objek yang ada didunia nyata, dimana dalam metode ini menggunakan alat bantu yaitu *UML (Unified Modeling Language)* yang terdiri dari : *Use Case Diagram, Object Diagram, Class Diagram,Sequence Diagram, Collaboration*

Diagram, Statechart Diagram Activity Diagram, Component Diagram, dan Deployment Diagram.

3.2.3.2. Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan adalah dengan menggunakan metode pendekatan *prototype. Prototyping* merupakan salah satu metode pengembangan perangat lunak yang banyak digunakan. *Prototype* dibangun untuk mendefinisikan kebutuhan.

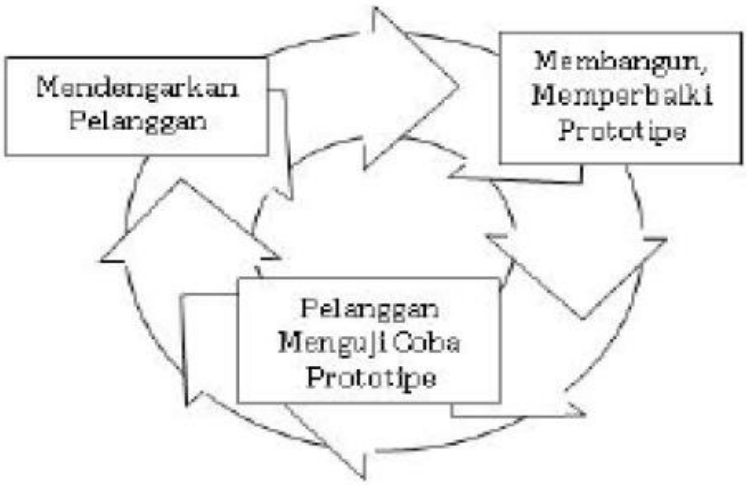
Prototyping merupakan salah satu metode pengembangan perangat lunak yang banyak digunakan. Dengan metode *prototyping* ini pengembang dan pelanggan dapat saling berinteraksi selama proses pembuatan sistem.

Sering terjadi seorang pengguna hanya mendefinisikan secara umum apa yang dikehendakinya tanpa menyebutkan secara detail *output* apa saja yang dibutuhkan, pemrosesan dan data-data apa saja yang dibutuhkan. Sebaliknya disisi pengembang kurang memperhatikan efesiensi algoritma, kemampuan sistem operasi dan *interface* yang menghubungkan manusia dan komputer.

Untuk mengatasi ketidak serasian antara pengguna dan pengembang ,maka harus dibutuhakan kerjasama yanga baik diantara keduanya sehingga pengembang akan mengetahui dengan benar apa yang diinginkan pelanggan dengan tidak

mengesampingkan segi-segi teknis dan pelanggan akan mengetahui proses-proses dalam menyelasaikan sistem yang diinginkan. Dengan demikian akan menghasilkan sistem sesuai dengan jadwal waktu penyelesaian yang telah ditentukan.

Kunci agar model *prototype* ini berhasil dengan baik adalah dengan mendefinisikan aturan-aturan main pada saat awal, yaitu pengguna dan pengembang harus setuju bahwa prototype dibangun untuk mendefinisikan kebutuhan. Prototype akan dihilangkan sebagian atau seluruhnya dan perangkat lunak aktual aktual direkayasa dengan kualitas dan implementasi yang sudah ditentukan .



Gambar 3.2 ***Paradigma Metode Pendekatan Prototype***

Paradigma

Sumber : Roger S. Pressman (2002)

Berikut adalah langkah-langkah dalam merancang sistem yang penulis gunakan dalam mekanisme pengembangan sistem dengan prototype , langkah-langkah tersebut adalah :

1. Mengidentifikasikan kebutuhan pemakai

   Analisis sistem mewarnai pemakai untuk mendapatkan gagasan dan apa yang diinginkan pemakai terhadap sistem.

2. Membangun *prototype*

   Analisis sistem mungkin bekerja sama dengan spesialis informasi lainya, menggunakan satu atau lebih peralatan prototype untuk mengembangkan sebuah prototype. Membangun *prototyping* dengan membuat perancangan sementara yang berfokus pada penyajian pada pelanggan (misalnya dengan membuat format *input* dan *output*)

3. Evaluasi Prototyping

   Evaluasi ini dilakukan oleh *user* apakah *prototyping* yang sudah dibangun sudah sesuai dengan keinginan pelanggan. Jika sudah sesuai maka akan diambil. Jika tidak *prototyping* direvisi dengan mengulang langkah 1, 2, dan 3.

3.2.3.3. Alat Bantu Analisis dan Perancangan

a. Use Case Diagram

Use-case diagram merupakan model diagram UML yang digunakan untuk menggambarkan *requirement* fungsional yang diharapkan dari sebuah sistem. *Use-case diagram* menekankan pada "siapa" melakukan "apa" dalam lingkungan sistem perangkat lunak yang akan dibangun. *Use-case diagram* sebenarnya terdiri dari dua bagian besar; yang pertama adalah *use case diagram* (termasuk gambar *use case dependencies*) dan u*se case description*.

Use-case diagram adalah gambaran *graphical* dari beberapa atau semua *actor*, *use-case*, dan interaksi diantara komponen-komponen tersebut yang memperkenalkan suatu sistem yang akan dibangun. *Use-case diagram* menjelaskan manfaat suatu sistem jika dilihat menurut pandangan orang yang berada di luar sistem. Diagram ini menunjukkan fungsionalitas suatu sistem atau kelas dan bagaimana sistem tersebut berinteraksi dengan dunia luar.

Use-case diagram dapat digunakan selama proses analisis untuk menangkap *requirement system* dan untuk memahami bagaimana sistem seharusnya bekerja. Selama tahap desain, *use-case diagram* berperan untuk menetapkan perilaku (*behavior*) sistem saat diimplementasikan. Kebutuhan atau

requirements system adalah fungsionalitas apa yang harus disediakan oleh sistem kemudian didokumentasikan pada model *use-case* yang menggambarkan fungsi sistem yang diharapkan (*use-case*), dan yang mengelilinginya (*actor*), serta hubungan antara actor dengan *use-case* (*use-case diagram*) itu sendiri.

b. Class Diagram

Class diagram digunakan untuk menampilkan kelas-kelas dan paket-paket di dalam system. *Class diagram* memberikan gambaran system secara statis dan relasi antar mereka. Biasanya, dibuat beberapa *class diagram* untuk system tunggal. Beberapa diagram akan menampilkan subset dari kelas-kelas dan relasinya. Dapat dibuat beberapa diagram sesuai dengan yang diinginkan untuk mendapatkan gambaran lengkap terhadap sistem yang dibangun.

Class diagram adalah alat perancangan terbaik untuk tim pengembang. Diagram tersebut membantu pengembang mendapatkan struktur system sebelum kode ditulis, dan membantu untuk memastikan desain terbaik dari system.

c. *Sequence* Diagram

Sequence diagram (diagram urutan) adalah suatu diagram yang memperlihatkan atau menampilkan interaksi-interaksi

antar objek di dalam sistem yang disusun pada sebuah urutan atau rangkaian waktu. Interaksi antar objek tersebut termasuk pengguna, display, dan sebagainya berupa pesan/message.

Sequence Diagram digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/even untuk menghasilkan output tertentu. *Sequence* Diagram diawali dari apa yang me-trigger aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan.

Diagram ini secara khusus berasosiasi dengan use case diagram. *Sequence* diagram juga memperlihatkan tahap demi tahap apa yang seharusnya terjadi untuk menghasilkan sesuatu didalam use case. *Sequence* diagram juga dapat merubah atribut atay method pada class yang telah dibentuk oleh class diagram, bahkan menciptakan sebuah class baru. *Sequence* diagram memodelkan aliran logika dalam sebuah system dalam cara yang visual.

d. Collaboration Diagram

Menggambarkan kolaborasi dinamis seperti *sequence* diagram. Dalam menjukkan pertukaran pesan, collaboration diagram menggambarkan object dan hubungannya (mengacu

ke konteks). Jika penekanannya pada waktu atau urutan gunakkan *sequence* diagram, tetapi jika penekanannya pada konteks gunakan collaboration diagram.

e. *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, di mana 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 menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses. Dipakai pada business modeling untuk memperlihatkan urutan aktifitas proses bisnis. *Activity* diagram dibuat berdasarkan sebuah atau beberapa use case pada use case diagram.

f. *Component* Diagram

Menggambarkan struktur fisik kode dari komponen. Komponen dapat berupa source code, komponen biner, atau executable *component*. Sebuah komponen berisi tenatang logic class atau class yang diimplementasikan sehingga membuat pemetaan dari logical view ke *component* view.

g. *Deployment* Diagram

Sebuah deployment diagram menunjukkan perangkat keras sistem dan perangkat lunak dalam perangkat keras tersebut. Diagram *Deployment* berguna ketika solusi perangkat lunak Anda dikerahkan di beberapa mesin dengan masing-masing memiliki konfigurasi yang unik.

Deployment Diagram mewakili pandangan pengembangan sistem sehingga akan hanya ada satu deployment diagram untuk satu sistem. *deployment* diagram terdiri dari node-node merupakan perangkat keras fisik yang digunakan untuk menyebarkan aplikasi. *deployment* diagram banyak di gunakan oleh *System Engineer*.

3.2.4. Pengujian Software

3.2.4.1 Pengujian *Black-box*

Menurut IEEE (1990) "Pengujian *Black-box* adalah pengujian yang dilakukan untuk mengevaluasi pemenuhan sistem atau komponen dengan kebutuhan fungsional tertentu."

Dari pengertian diatas penulis menyimpulkan bahwa pengujian *black-box* hanya berfokus terhadap kebutuhan fungsionalitas seperti tombol input data, simpan data dan tampilan lainnya tanpa mengetahui logika yang ada di dalamnya.

Pengujian *Black Box* berusaha menemukan kesalahan dalam kategori sebagai berikut :

1. Fungsi-fungsi yang tidak benar atau hilang.
2. Kesalahan interface.
3. Kesalahan dalam akses data base eksternal.
4. Kesalahan kinerja.
5. Inisialisasi dan kesalahan terminasi.

3.3. Analisis Sistem Yang Berjalan

3.3.1. Analisis Prosedur Yang Sedang Berjalan

Analisis prosedur merupakan kegiatan dari tahapan-tahapan yang menerangkan mengenai proses yang sedang berjalan dan dokumen apa saja yang telibat. Analisis prosedur yang berjalan ini menggambarkan

analisis sistem yang berjalan di TK. Berikut ini merupakan prosedurnya :

1. Guru memberikan materi

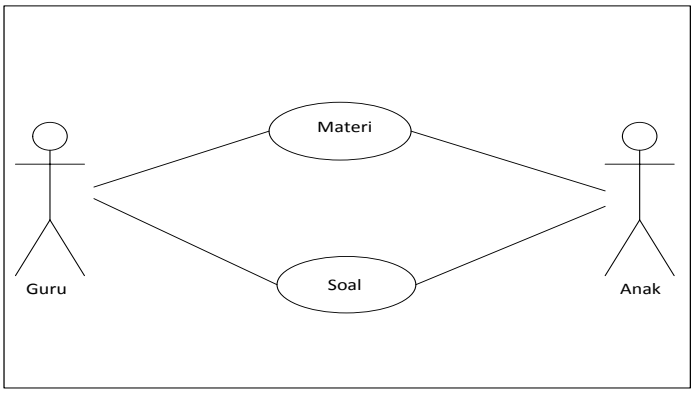
2. Anak menyimak materi yang diberikan oleh guru

3. Guru memberikan soal

4. Anak menjawab

3.3.1.1. Use case yang sedang berjalan

Berikut ini menggambarkan use cse yang sedang berjalan saat ini



Gambar 3.3 Use case sistem yang sedang berjalan

1. Identifikasi aktor pada sistem yang sedang berjalan

Tabel 3.1 identifikasi aktor pada sistem yang sedang berjalan

NO	Use Case	Deskripsi
1.	Anak	Aktor yanng menerima materi dan soal.
2	Guru	Aktor yang memberi materi dan soal.

2. Identifikasi use case pada sistem yang sedang berjalan

Tabel 3.2 Identifikasi use case pada sistem yang sedang berjalan

NO	Use Case	Deskripsi
1.	Materi	Proses dimana materi diberikan oleh guru dan diterima olek anak.
2	Soal	Proses dimana soal diberikan oleh guru dan diterima olek anak.

1. Skenario Use case Materi

No Use Case : 01

Nama Use Case : Materi

Aktor : Guru dan Anak

Tujuan : Untuk memberikan dan menerima materi

Skenario

Tabel 3.3. Skenario Use Case Sistem Yang Sedang berjalan no.1

Guru	Anak
1. Guru memberikan materi belajar	2. Anak menerima materi yang diberian oleh guru

2. Skenario Use case

No Use Case : 02

Nama Use Case : Soal

Aktor : Guru dan Anak

Tujuan : Untuk memberikan dan menerima soal

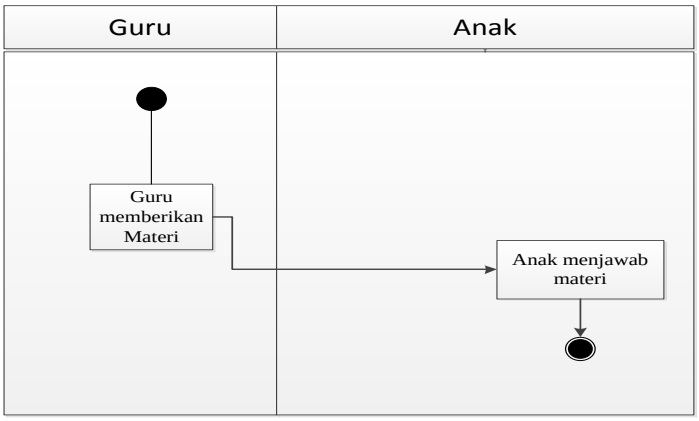
Skenario

Tabel 3.4. Skenario Use Case Sistem Yang Sedang Berjalan no.2

Guru	Anak
3. Guru memberikan materi belajar	4. Anak menerima materi yang diberian oleh guru

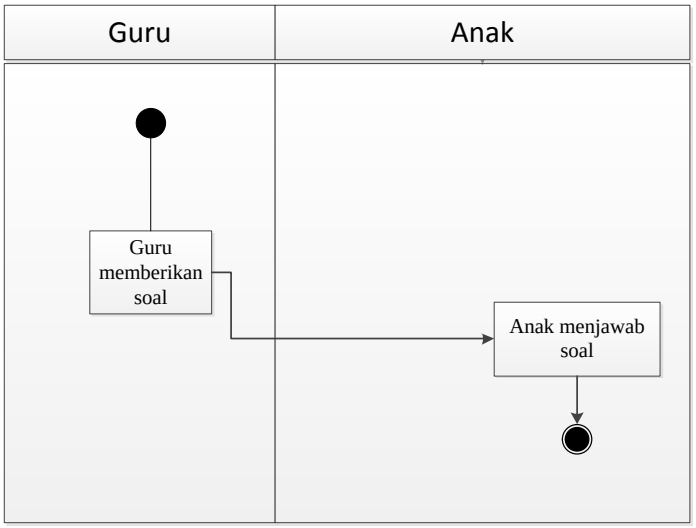
3.3.1.2. Activity diagram sistem yang sedang berjalan

1. Aktivity Diagram materi pada sistem yang sedang berjalan.



Gambar 3.4 Aktivity diagram materi pada sistem yang sedang berjalan

2. Aktivity diagram soal pada sistem yang sedang berjalan



Gambar 3.5 Aktivity diagram soal pada sistem yang sedang berjalan

3.3.2. Evaluasi Sistem yang sedang berjalan

Setelah melakukan analisis terhadap sistem yang berjalan. Dan juga seperti yang sudah dibahas pada identifikasi masalah, berikut hasil evaluasinya :

Tabel 3.5 Evaluasi system yang berjalan

NO	Permasalahan	Pemecahan
1	Dengan sistem belajar yang ada saat ini anak akan	Merancang aplikasi edukasi yang diharapakan dapat lebih menyenangkan agar minat belajar pada anak lebih besar.

	lebih cepat bosan dalam belajar	
2.	Kurang dimanfaatkanya teknologi yang semakin maju sebagai bagian dari proses belajar	Dengan dirancangnya aplikasi edukasi berbasis mobile ini akan memberikan alternatif anak dalam belajar, aplikasi ini juga membuat anak dapat belajar dimana saja dan kapan saja.