**BAB III**

**PERANCANGAN PERANCANGAN APLIKASI PEMBERITAHUAN BERBASIS ANDROID**

Berdasarkan tinjauan literatur dan landasan teori yang telah dibahas pada bab sebelumnya, pada bagian ini penulis akan menguraikan pembahasan tentang sistem kerja aplikasi serta perancangan aplikasi pemberitahuan dengan Firebase. Untuk itu penulis membagi bab ini menjadi beberapa sub bab yang terdiri dari Analisi kerja Aplikasi, fitur *Firebase* *Authentication* , *Firebase Cloud Messaging*, *Firebase Realtime Database*, *Firebase Cloud storage* yang digunakan pada aplikasi.

1. **Analisis Kerja Aplikasi**

Perancangan aplikasi sistem pemberitahuan ini sebenarnya mempunyai beberapa tujuan yang menguntungkan baik pada pihak pemakai maupun penyedia. Keuntungan diantara lain :

1. Bagi pemakai
2. Proses penerimaan informasi menjadi lebih cepat karena hanya perlu melihat notifikasi dari aplikasi. Sehingga pengguna tidak perlu mencari in*form*asi pada papan pengumuman.
3. Mempermudah pemakai terutama orang tua murid untuk mengawasi perkembagan pelajaran anak.
4. Mempermudah pemakai terutama orang tua murid untuk melakukan tanggapan mengenai suatu pemberitahuan atau berita dari sekolah dengan fitur komentar yang disediakan.
5. Bagi penyedia
6. Proses penyampaian informasi sangat efisien karena tidak perlu media fisik seperti kertas.
7. Mengurangi jumlah pekerjaan fisik bagi pengurus karena hanya perlu mengetik pada aplikasi.
8. Mempermudah penyedia untuk melayani tanggapan yang diberikan oleh orang tua murid terhadap informasi atau pemberitahuan yang disampaikan.

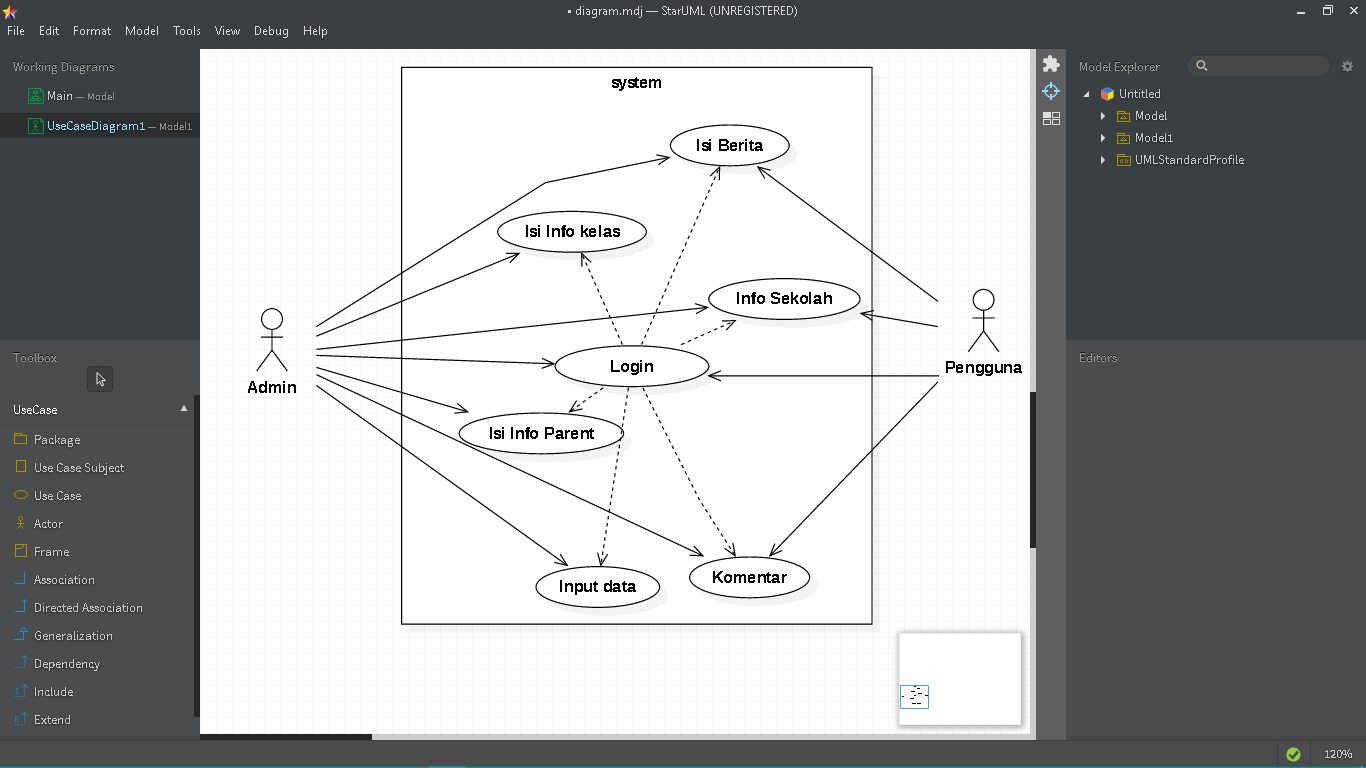
Aplikasi pemberitahuan berkerja pada *Operating System* android sehingga fitur disesuaikan dengan kebiasaan penggunaan android, aplikasi dimulai dengan tampilan *login* *form* yang berisi *textbox* nomor telepon yang harus diisi dengan nomor telepon yang telah teregistrasi sebelumnya kemudian ketika *User* menekan tombol *login* maka akan muncul sebuah *popup menu,* kemudian akan ada otentifikasi dari firebase untuk verifikasi kode masuk yang dikirimkan lewat pesan singkat *Short Message Service* (SMS) yang berupa kombinasi 6 buah angka desimal.

Untuk lebih memperjelas alur proses program yang dilakukan pada aplikasi pemberitahuan ini, maka penjelasan yang lebih lanjut akan ditampilkan pada diagram *Unified Modelling Language* (UML) dalam beberapa jenis diagram.

1. Diagram *Use Case*

**GAMBAR 3.1**

DIAGRAM USE CASE



*Sumber: Data Olahan (2018)*

Diagram *Use Case* memberikan gambaran mengenai sistem yang akan dirancang dengan menampilkan aktor-aktor yang terlibat dalam masing-masing proses.

1. Diagram Aktivitas

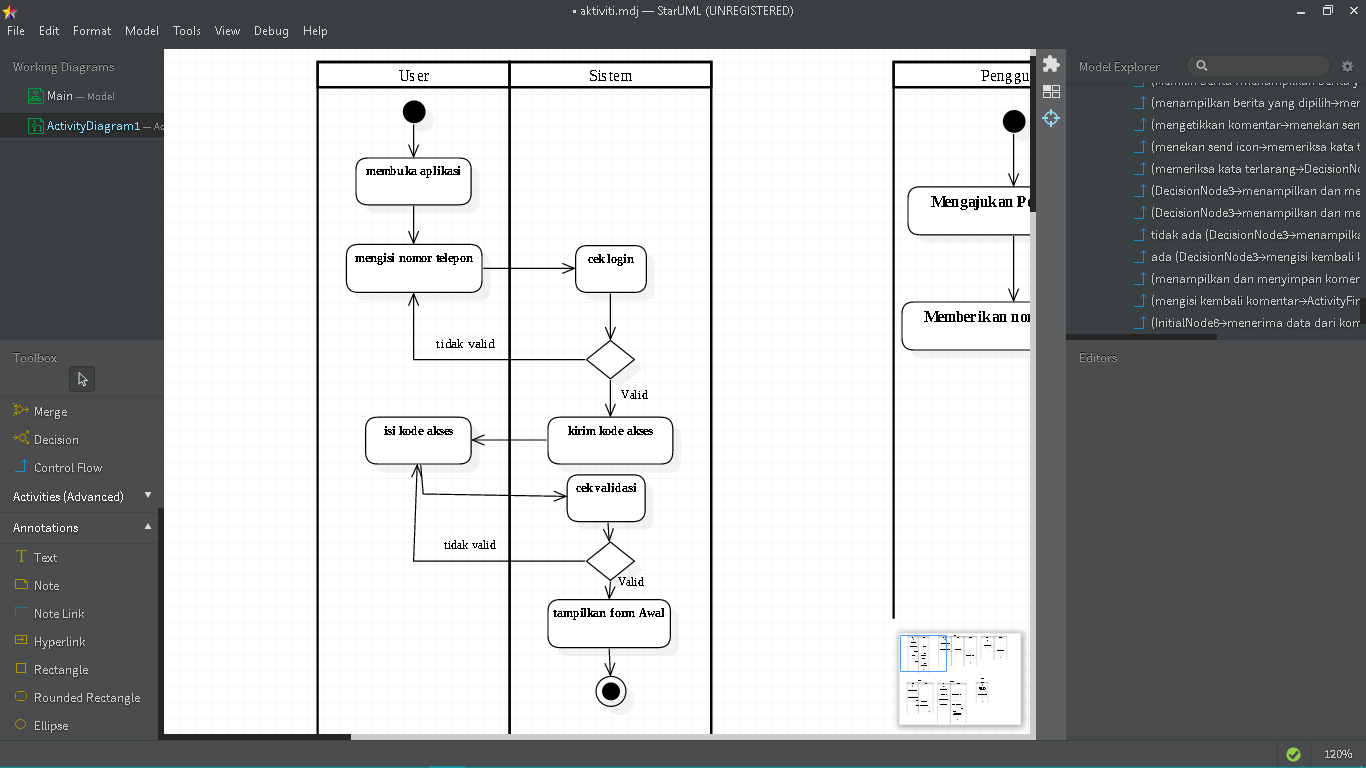
Diagram aktivitas memberikan gambaran mengenai berbagai aliran aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alur berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. Pada diagram ini akan diberikan gambaran alur proses berdasarkan masing-masing aktivitas yang terjadi pada sistem.

Berikut adalah beberapa diagram aktivitas pada sistem ini.

1. Diagram aktivitas *login*

**GAMBAR 3.2**

DIAGRAM AKTIVITAS *LOGIN*



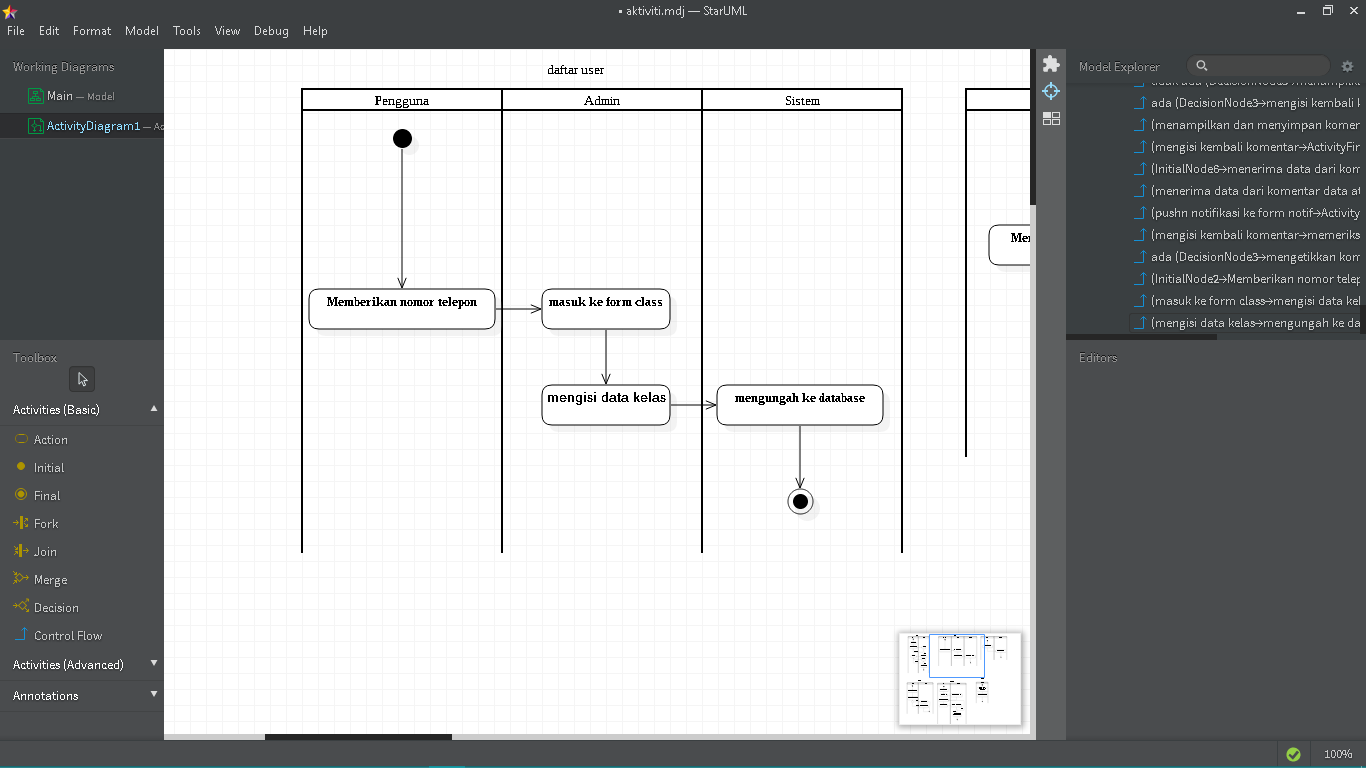
*Sumber: Data Olahan (2018)*

Tahap pertama sebelum *User* bisa menggunakan aplikasi ini adalah dengan melakukan *login* terlebih dahulu. Tahap ini berfungsi untuk mencegah pihak yang tidak berkepentingan mengakses data-data yang ada pada program. Cara *login* di sini, yaitu *User* akan diminta untuk mengisi nomor teleponyang sudah didaftarkan pada *database*  kemudian *Firebase Authentication* mengirimkan kode akses ke nomor telepon yang terdaftar untuk validasi.

1. Diagram aktivitas pendaftaran *User*

**GAMBAR 3.3**

DIAGRAM AKTIVITAS PENDAFTARAN PARENT DAN KELAS



*Sumber: Data Olahan (2018)*

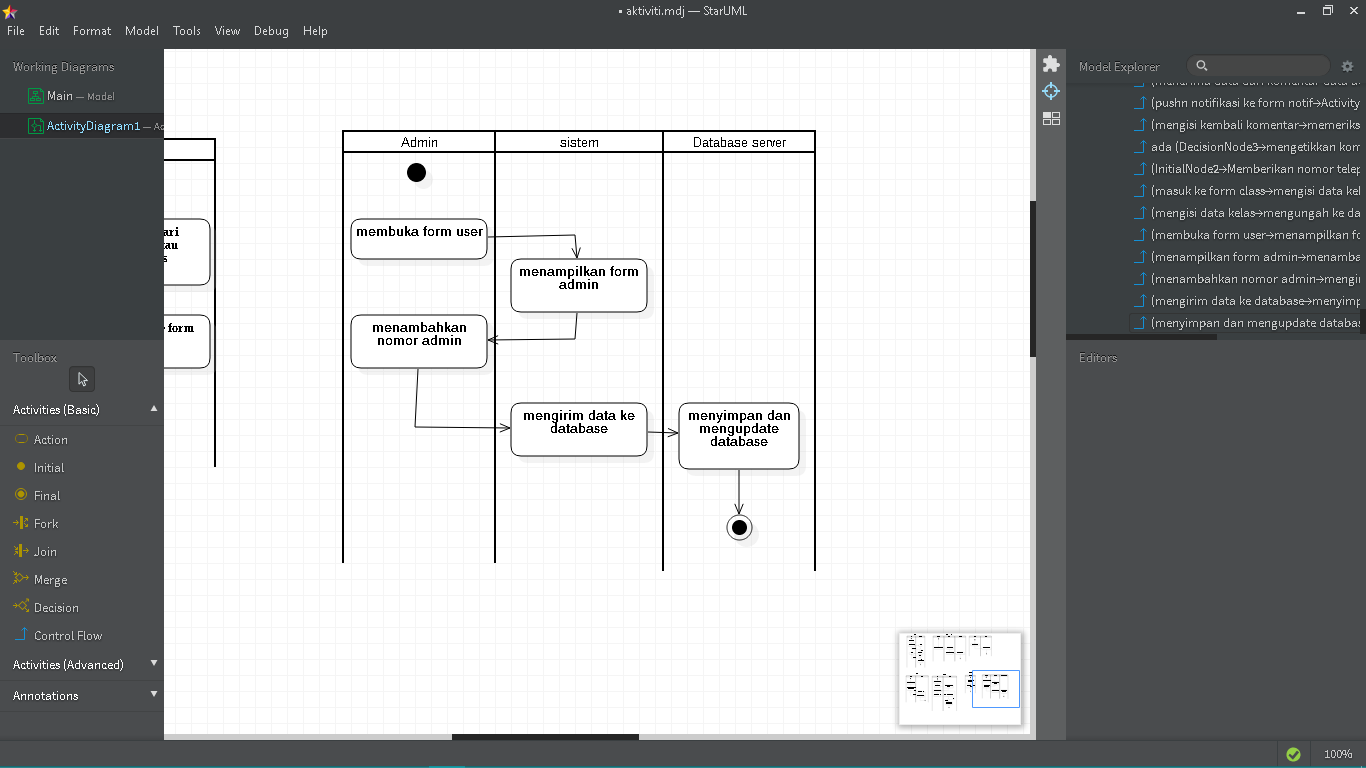
Untuk melakukan pendaftaran kelas dipelukan *input* data kelas yang terdiri dari *total child, teacher, mentor1, mentor2* yang telah tentukan oleh pengurus sekolah *form* ini juga dapat melakukan *edit* dan hapus data dengan memilih langsung kelas yang akan diubah.

Admin harus menambahkan nama anak dan orang tua pada *from Class* Kemudian melakukan *input* data *User* sebagai *parent.* Pada *form class* ketika akan melakukan pengisian data kelas maka semua nomor telepon orang tua yang menjadi wali anak akan langsung menjadi pengguna aplikasi, tidak hanya bisa melakukan pendaftaran, namun bisa melakukan penghapusan ataupun peng-*edit-*an pada *User* yang sudah ada sebelumnya.

1. Diagram aktifitas Menambah *User* admin

**GAMBAR 3.4**

DIAGRAM AKTIVITAS PENDAFTARAN ADMIN

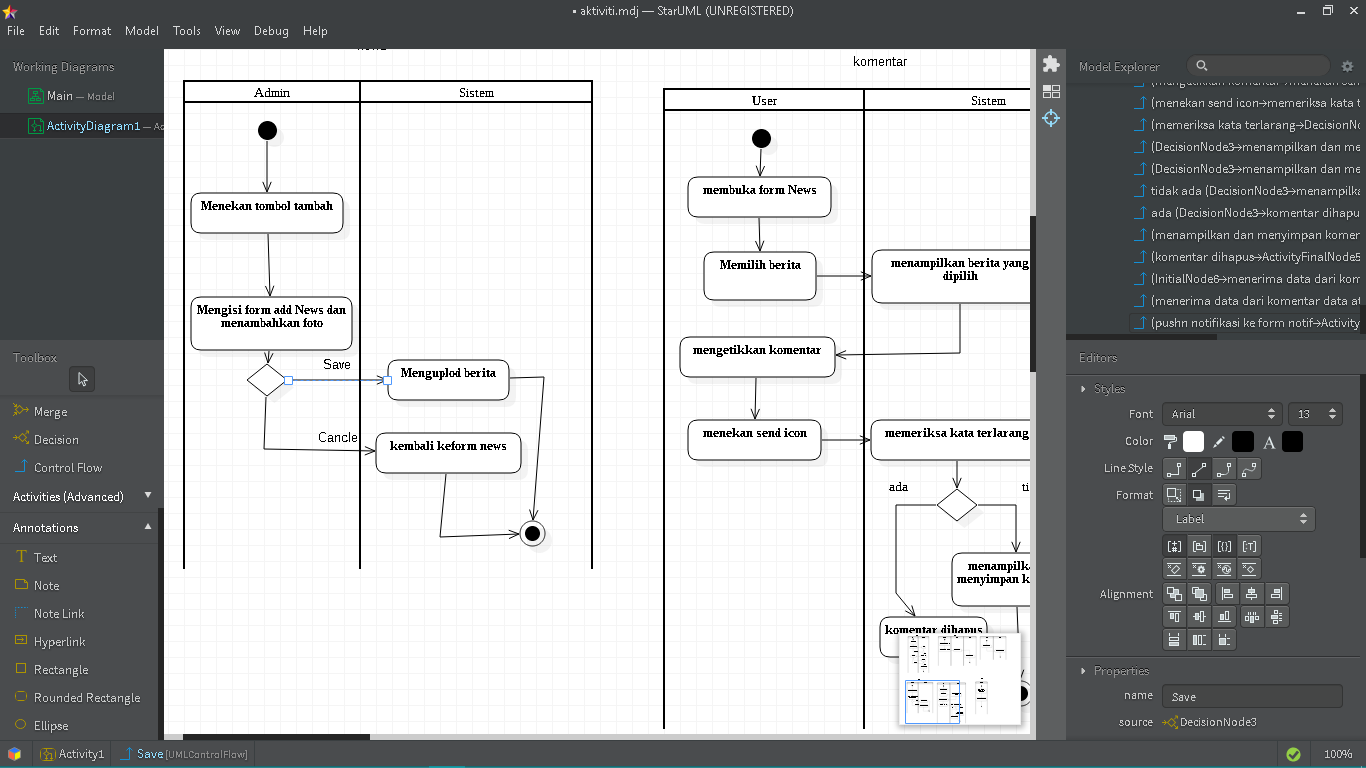


*Sumber: Data Olahan (2018)*

1. Diagram aktivitas Menambah Berita

**GAMBAR 3.5**

DIAGRAM AKTIVITAS MENAMBAH BERITA

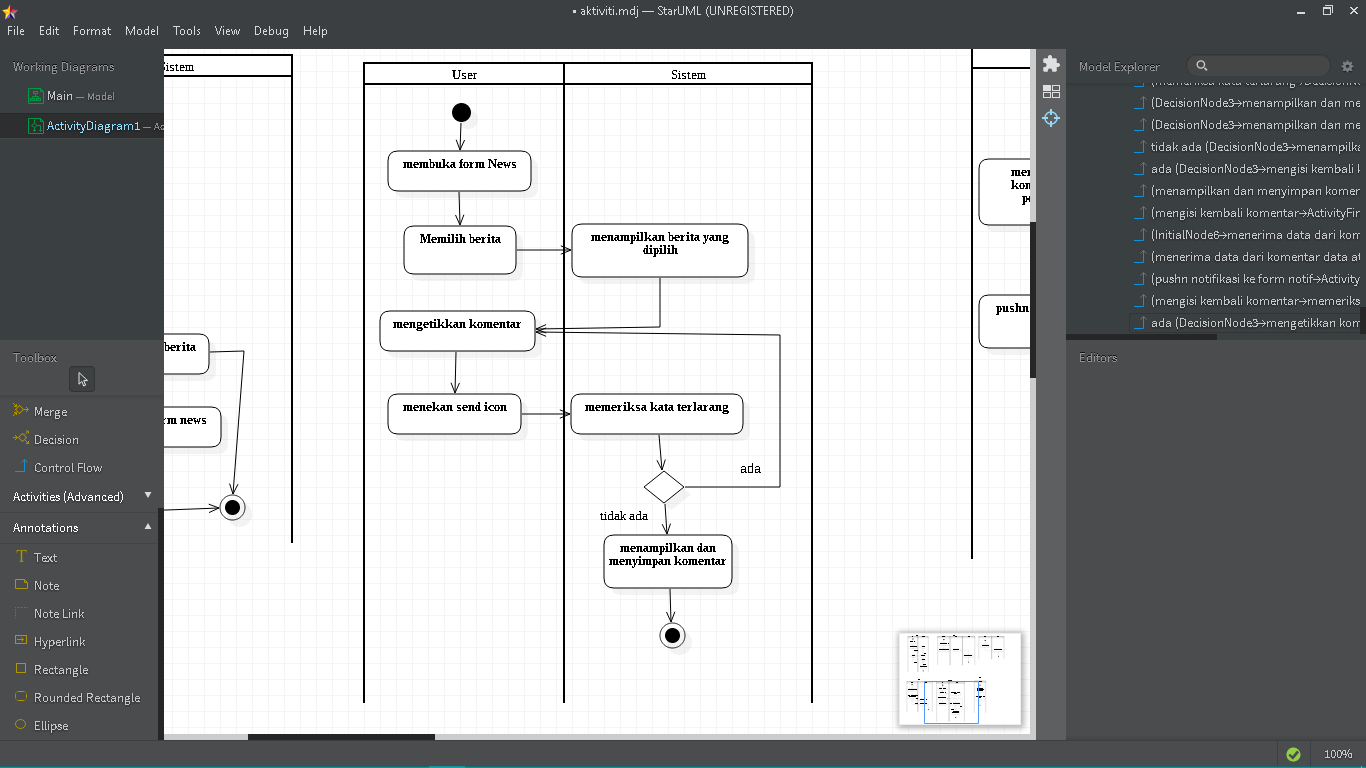
  
*Sumber: Data Olahan (2018)*

Pegisian berita hanya dapat dilakukan oleh admin dan di-*upload* oleh admin dengan menekan tombol tambah “+” pada bagian kanan atas tampilan program kemudian akan tampil *form* *add news* sehingga admin dapat mengisi pada box yang disediakan yaitu kolom *body* yaitu untuk mengisi berita dengan tipe data *string. Form* ini juga dapat mengisikan gambar berjenis Jpeg dan png.

1. Diagram aktivitas komentar

**GAMBAR 3.6**

DIAGRAM AKTIVITAS KOMENTAR



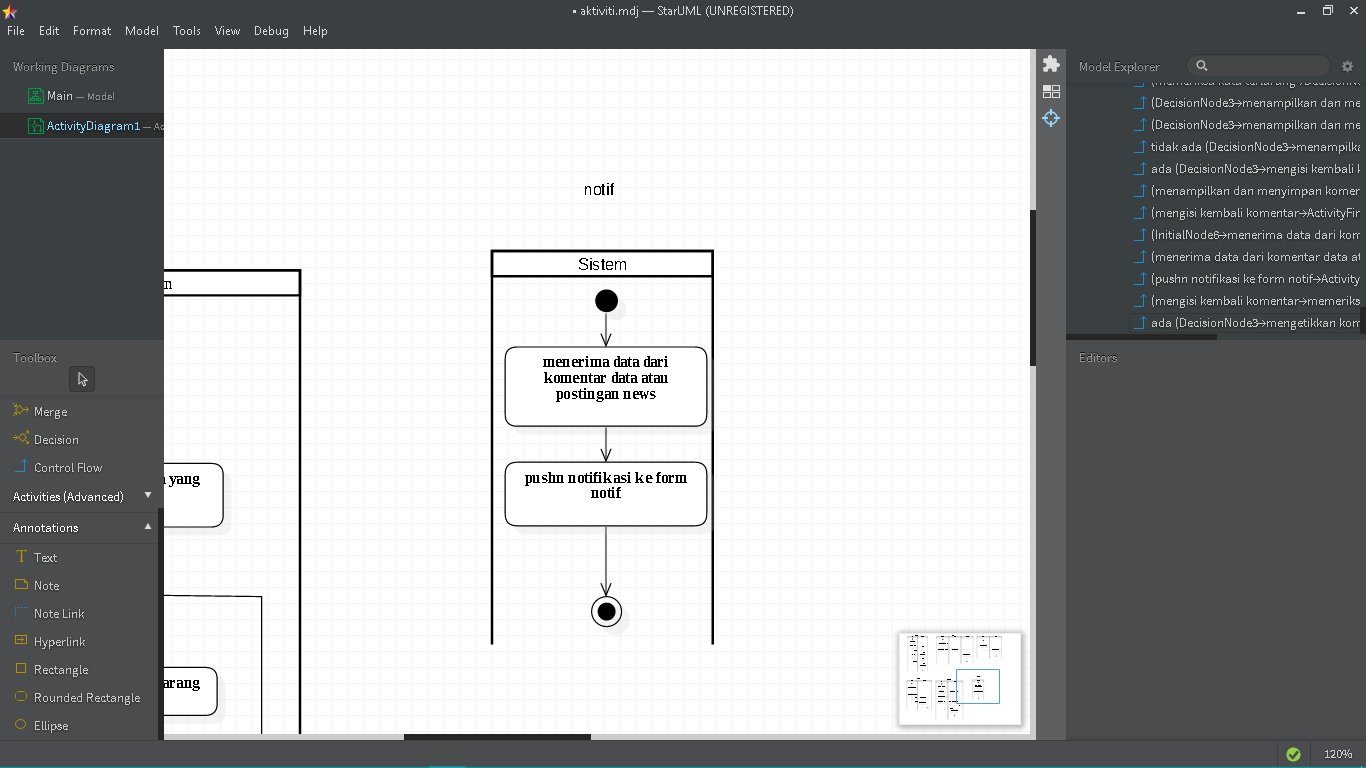
*Sumber: Data Olahan (2018)*

Untuk melakukan komentar pada berita yang dipilih oleh semua *User* akan tampil sebuah kolom isi komentar pada bagian paling bawah layar sehingga *User* dapat langsung mengetikan komentar pada kolom tersebut kemudian menekan tombol dengan icon (>) kirim dikanannya. Setelah *User* menekan tombol kirim maka kalimat yang dimasukkan oleh *User* akan langsung dicek pada sistem apakah terdapat bahasa yang kasar atau yang tidak diizinkan, jika ada bahasa yang tidak dizinkan maka komentar yang dikirim akan dikembalikan ke kolom komentar untuk di-*edit* oleh *User*. Sementara jika tidak terdapat bahasa yang tidak dizinkan maka komentar *User* akan segera di posting ke *form* komentar dan disimpan ke dalam database.

1. Diagram aktivitas *Notification*

**GAMBAR 3.7**

DIAGRAM AKTIVITAS NOTIFICATION

  
*Sumber: Data Olahan (2018)*

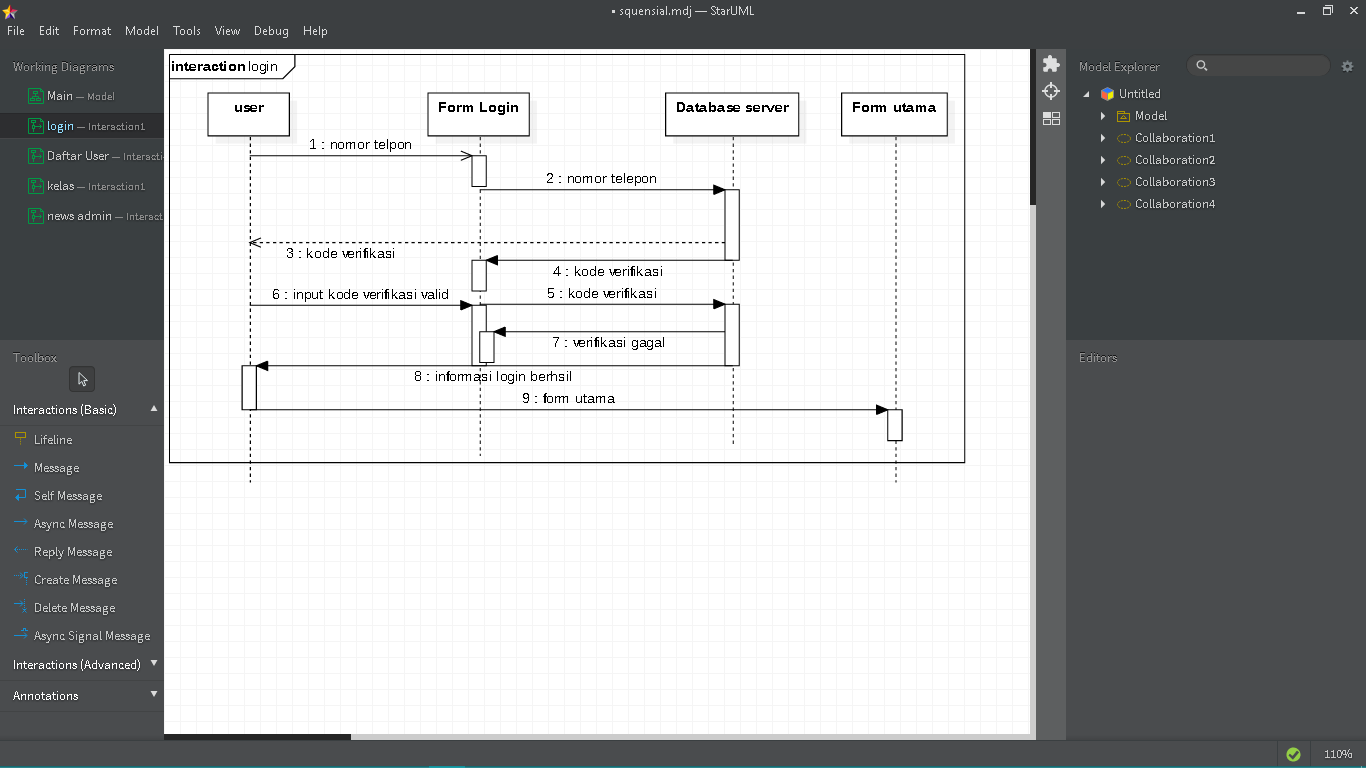
Notifikasi dilakukan pada system dimana ketika komentar ataupun berita yang telah diposting oleh *User* lain akan segera dikirimkan push notificationnya sehingga semua *User* yang berhubungan dengan tipe pemberitahuannya mendapatkan notifikasi pada *form* *notif*. *User* dapat menekan kolom notif yang telah masuk dan segera mambuka notifikasi tersebut.

1. Diagram Sekuensial

Diagram sekuensial memberikan gambaran mengenai proses berjalannya sistem berdasarkan urutan kerja. Berikut adalah diagram sekuensial yang menggambarkan kerja sistem yang akan dibangun.

1. Diagram sekuensial *login*

**GAMBAR 3.10**

DIAGRAM SEKUENSIAL *LOGIN*  


*Sumber: Data Olahan (2018)*

Hal pertama yang dilakukan ketika membuka aplikasi adalah proses *login.* Hal ini bertujuan untuk mencegah orang yang tidak berkepentingan untuk mengakses aplikasi. Setelah membuka aplikasi, maka pengguna akan dihadapkan pada tampilan *form login*  pengguna diminta untuk memasukan nomor telepon yang telah terdaftar sebelumnya oleh *User* Admin pada aplikas. Nomor telepon yang dimasukkan tanpa regional kode misalnya : +6289661649693 hanya perlu memasukkan 89661649693 (tanpa kode regional).

Setelah melakukan pengisian nomor telepon*,* maka akan dilakukan pengecekkan pada *database server,* kolom verifikasi akan tampil setelah pengecekkan nomor telepon pada database *database* sudah sesuai, kemudian pengguna diminta untuk memasukkan kode verifikasi yang dikirimkan melelui SMS yang terdiri dari enam kombinasi angka desimal kode verifikasi ini berasal dari Firebase *Authentication* pada setiap aktifitas *login* *User* pada aplikasi. Jika nomor telepon yang dimasukkan oleh *User* tidak terdapat pada database *server* maka *login* akan ditolak.

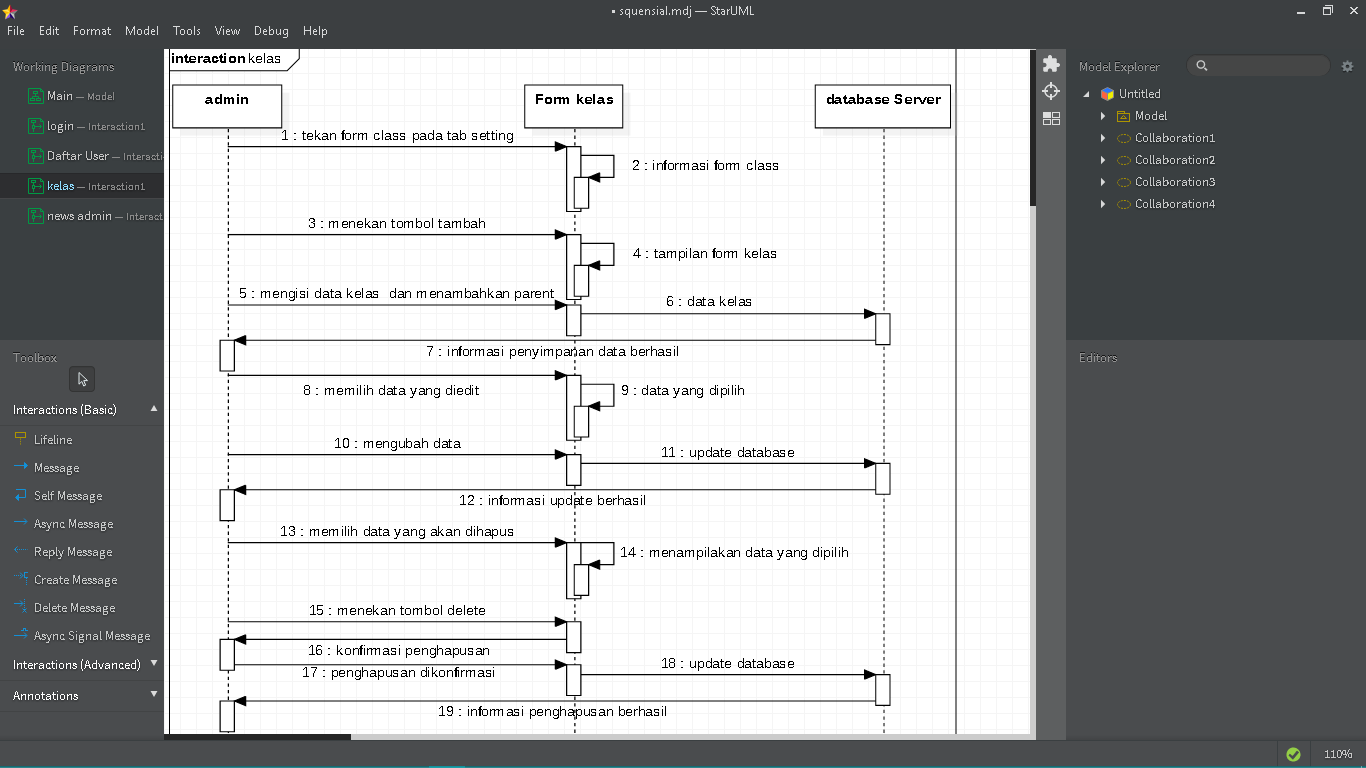
Pada verifikasi *login* juga akan dilakukan pengecekan pada database  *server* jika kode yang dimasukkan oleh *User* tidak sama dengan angka yang dikirimkanoleh Firebase *Autentication* maka *login* akan ditolak Proses verifikasi nomor telepon dapat membutuhkan waktu yang bervariasi tergantung kecepatan koneksi internet pada perangkat yang digunakan untuk menggunakan aplikasi.

Setelah melakukan pengisian kode verifikasi dan verifikasi kode telah benar, maka aplikasi akan menampilkan *form* utama yaitu *form* *news*. Pengisian kode verifikasi sendiri dapat secara otomatis atau manual, jika nomor telepon yang digunakan pada perangkat sama dengan nomor yang didaftarkan sebagai *User* maka system akan mengisi kode verifikasi secara otomatis. Sedangkan jika nomor yang digunakan pada perangkat yang menggunakan aplikasi berbeda maka *User* harus mengisikan kode secara manual.

1. Diagram sekuensial pendaftaran kelas dan *parent*

**GAMBAR 3.11**

DIAGRAM SEKUENSIAL PENDAFTARAN KELAS DAN PARENT



*Sumber: Data Olahan (2018)*

Pada bagian ini, *User* admindapat melakukan pengaturan *User* parent berupa penambahan, peng-*edit*-an, dan penghapusan *User.* Cara mengaksesnya adalah dengan membuka *form class* pada *tab setting,* admin harus mengisikan nomor telepon dan nama orang tua murid yang akan menjadi *User parent* pada tombol *add* yang ada dibawah in*form*asi kelas*,* setelah *add* ditekan akan muncul sebuah *form Add Parent*.Setelah mengisikan data *User* baru akan tersimpan dengan menekan *icon* *save* pada kanan atas *form* pengisian data. Data yang tersimpan dapat dihapus dan edit sewaktu-waktu oleh admin adapun fungsi dari data diri member ini selain bisa sebagai sarana keamanan bagi para pengguna dari admin maupun orang tua murid.

Jika *field* *Add Parent* ada yang kosong maka ada informasi kesalahan yang memberitahukan kepada *User* bahwa *User* tidak dapat menyimpan data *blank.*

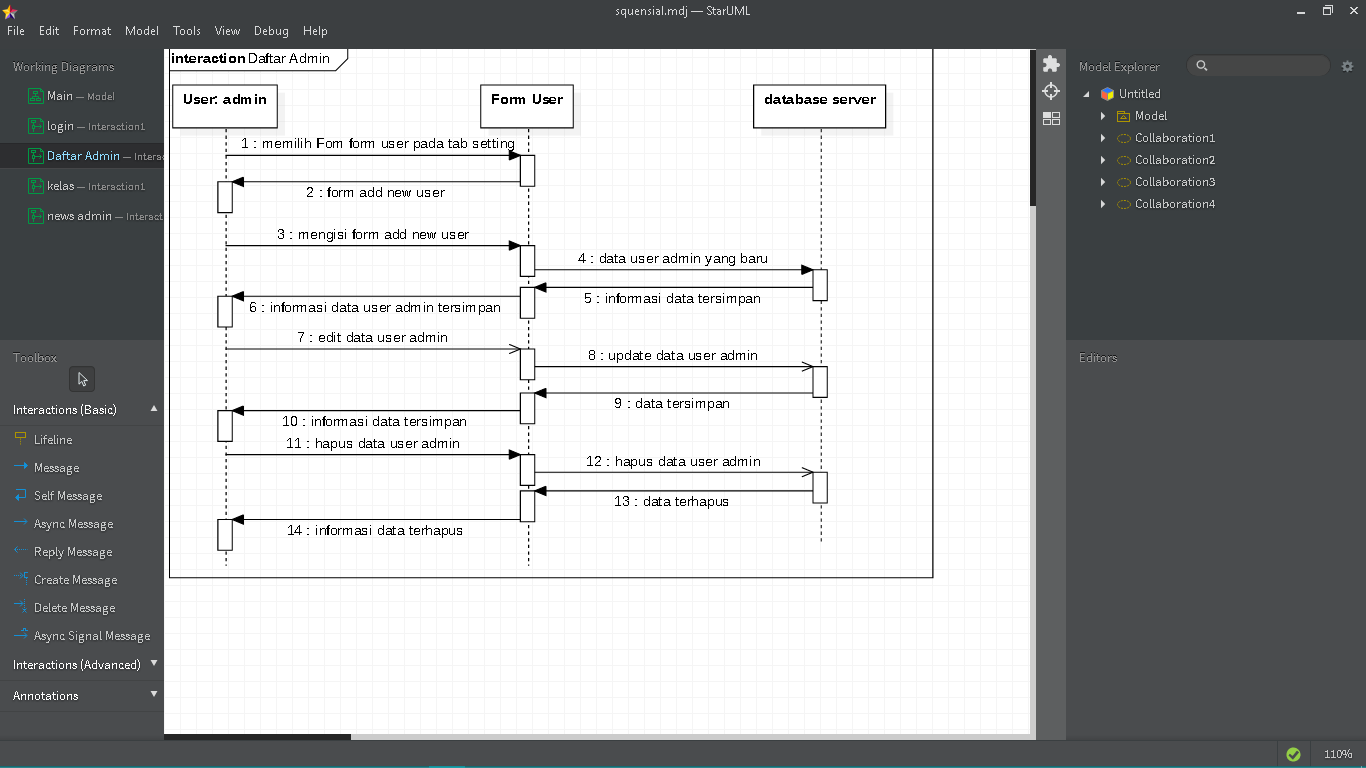
Untuk bagian penghapusan dan peng-*edit-*an, adminhanya perlu memilih data *User* mana yang akan dihapus atau *edit,* kemudian *klik* pada *Button* yang telah disediakan, dan melakukan konfirmasi pada saat sudah selesai dengan memilih “ok” pada *messagebox* yang muncul.

*User* parent yang telah terhapus dari datar kelas harus melakukan logout agar tidak terjadi kesalahan pada system. *System* tidak dapat secara otomatis memaksa *User* keluar dari aplikasi, sekalipun *database server* telah ter-*update* dan informasi *User* telah dihapus.

1. Diagram sekuensial tambah Admin

**GAMBAR 3.12**

DIAGRAM SEKUENSIAL PENAMBAHAN *USER* ADMIN



*Sumber: Data Olahan (2018)*

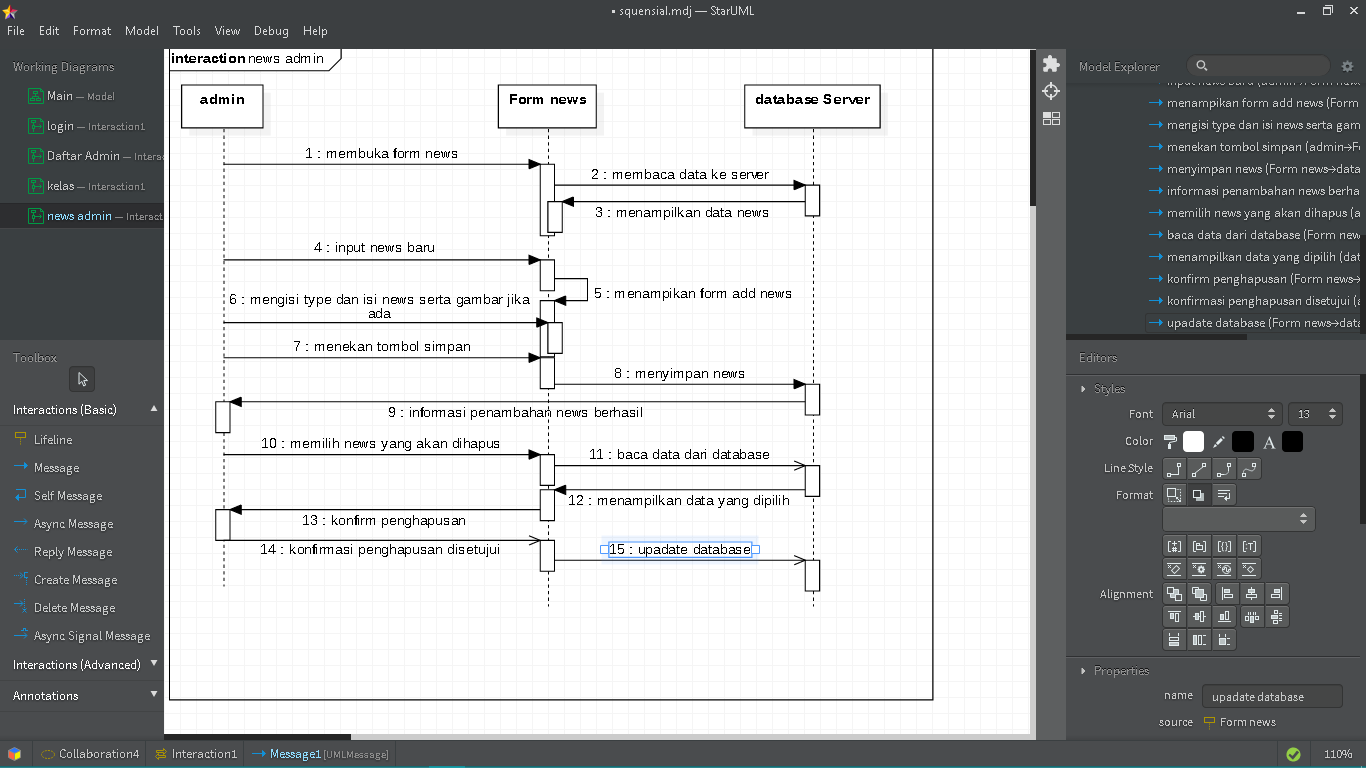
Pada bagian pendaftaran admin terdapat pada tab *setting user.* Untuk menambahkan *User* admin baru diperlukan *login* *User* admin sebelumnya. Pada *form* admin *user* menekan tombol “+” pada bagian kanan atas *form*, setelah itu akan tampil sebuah *form* pengisian nomor telepon yang akan digunakan sebagai admin dan nama admin yang akan dibuat.

Setelah mengisikan nomor telepon dan nama admin selanjutnya menekan tombol *icon save* yang ada di kanan atas *form*. Jika *form* tidak terisi dengan benar makan informasi kesalahan akan muncul dan mengharuskan mengisi semua kolom isian.

1. Diagram sekuensial *Post News*

**GAMBAR 3.13**

DIAGRAM SEKUENSIAL POST NEWS



*Sumber: Data Olahan (2018)*

Pada saat selesai *login* pada *user* admin akan segera ditampikan *form* news sehingga admin dapat segera mengisi pemberitahuan atau berita yang akan dipasang.

Untuk menambahkan pemberitahuan atau berita admin menekan tombol dengan *icon* “+” terlebih dahulu, kemudian akan tampil sebuah *form* pengisian berita. Pada *form* pengisian berita terdapat *field type* yang berfungsi untuk memisahkan berita yang di-*posting* kepada *user* *parent* yang terdaftar pada kelas masing-masing pada *field type,* jika ditekan akan tampil *list* kelas yang terdaftar dan juga “All” yang merupakan tipe untuk semua kelas atau informasi yang dipasang merupakan informasi umum yang perlu dilihat oleh semua *parent* yang terdaftar.

Pada *field* *post news* admin dapat mengisikan rikasan atau rincian berita yang akan dipasang. Tombol pemilihan gambar berada di bawah *field post news* yang berfungsi untuk mengunggah gambar pada *posting*-an. Jika berita yang akan disampaikan terdapat gambar maka admin harus telah memiliki gambar yang akan ditampilkan pada penyimpanan perangkat yang digunakan. Pada sistem ini tidak dapat langsung mengambil gambar menggunakan kamera perangkat.

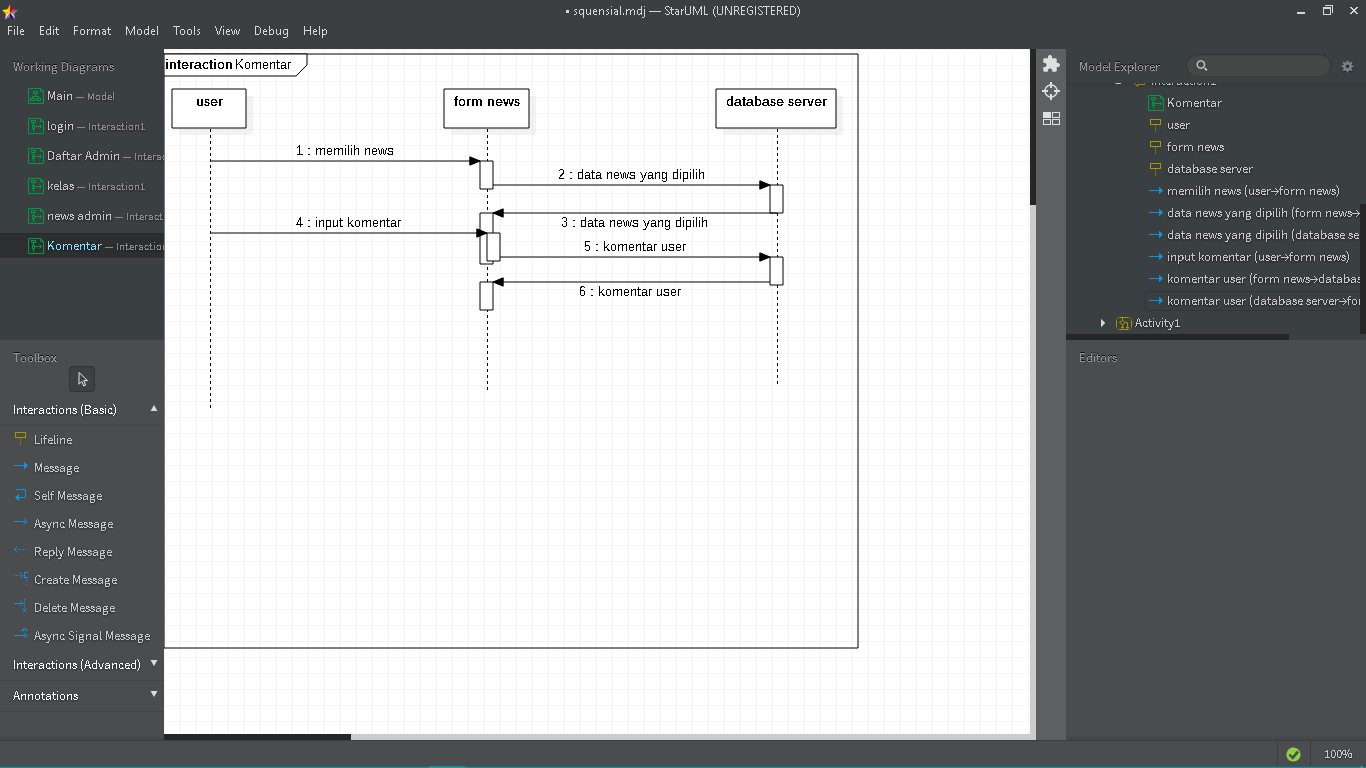
Setelah pengisian berita maka admin perlu menekan tombol dengn *icon save* untuk konfirmasi penambahan berita. Setelah konfirmasi maka *form news* akan ditampilkan kembali.

Pada *post news user* juga tidak dapat menyimpan *field* kosong sehingga sistem akan memaksa *user* untuk mengisi data atau keluar dari pengisian.

1. Diagram sekuensialkomentar

**GAMBAR 3.14**

DIAGRAM SEKUENSIAL KOMENTAR



*Sumber: Data Olahan (2018)*

Pada setiap berita yang telah di-*posting* oleh admin, semua *user* yang terdaftar pada kelas yang sesuai dengan *type* berita yang di-*posting* dapat melakukan komentar, komentar dari semua *User* akan di-*filter* terlebih dahulu jika terdapat kata atau suku kata yang tidak diperkenankan untuk digunakan serta dapat menyinggung perasaan orang lain maka komentar *user* tidak akan dikirim.

1. Perancangan *Database Server*

Dalam perancangan sistem basis data untuk aplikasi pemberitahuan ini menggunakan Firebase realtime database NoSQL yang memiliki struktur data base yang berbeda dengan *database* SQL. Pada database firebase bebentuk seperti sebuah pohon jaringan, setiap *object* yang di-*input* memiliki *property.* Setelah object dibuat firebase akan secara otomatis men-*generate* sebuah *id*. Berikut ini adalah rincian *object* yang ada pada *database*:

1. User

Terdiri dari *property* antara lain: childname, classid, createdat, createdby, phonenumber, typeUser, Username.

1. Admin

Terdiri dari *property* antara lain: phonenumber dan Username.

1. Class

Terdiri dari *property* antara lain: classid, classname, createdat, createdby, mentor1, mentor2, teacher, totalchild.

1. Class parent

Terdiri dari *property* antara lain: classid, createdat, createdby, terdapat sub object parent pada *object* ini yang terdiri: childname, classid, createdat, createdby, phonenumber, typeUser, Username.

1. Newsadmin

Firebase akan men-*generate id* untuk setiap *news* dibuat dan di-posting oleh admin *property* newsadmin terdiri dari : bodynews, classid, createdat, createdby, idx, typenews, urlimage

1. Newsparent

*Object* newsparent ini sama dengan newsadmin hanya perbedaannya pada isi dari typenews. *Property* newsparent terdiri dari : bodynews, classid, createdat, createdby, idx, typenews, urlimage

1. Comment

*Object* komentar Terdiri dari *property* antara lain: bodycomment, createdat, createdby, idnews,idx.

1. Notif

Tediri dari *property :* bodynews, createdat, createdby, idnews, idx, notifbody, urlimage.

1. Notifadmin

Tediri dari *property :* bodynews, createdat, createdby, idnews, idx, notifbody, urlimage.

1. Notifall

Tediri dari *property :* bodynews, createdat, createdby, idnews, idx, notifbody, urlimage.

1. Notifparent

Tediri dari *property :* bodynews, createdat, createdby, idnews, idx, notifbody, urlimage.

1. **Perancangan Tampilan Antarmuka**

Sebagai suatu aplikasi yang nantinya akan ditujukan untuk digunakan pada suatu intansi, maka diperlukan perancangan antarmuka agar *User* dapat memahami gambaran umum aplikasi yang akan dipakai. Berikut beberapa perancangan antarmuka yang akan digunakan :

1. Perancangan Tampilan Halaman *login*

**GAMBAR 3.15**

PERANCANGAN TAMPILAN HALAMAN LOGIN

**A**

**+6289-999-999-999**

*Sumber: Data Olahan (2018)*

Pada saat aplikasi pertama dijalankan, maka akan muncul tampilan *form* utama yang disini ada beberapa menu namun hanya ada satu menu yang bisa dipilih, yaitu menu *login.* Sebab sebelum *login,* fungsi lain tidak akan bisa diakses. Berikut adalah penjelasan dari tampilan antarmuka dari gambar di atas: pada gambar paling atas merupakan logo instansi, *textbox* di bawah logo diisi dengan nomor telepon, sedangkan tombol *login* (A) dibawah *textbox* untuk melakukan *login*.

1. Perancangan Tampilan *Form verify*

**GAMBAR 3.16**

PERANCANGAN TAMPILAN VERIFIKASI

**B**

**999999**

*Sumber: Data Olahan (2018)*

Setelah *login* *form* terisi dan data yang dimasukan sudah benar maka akan tampil *form* *verify* yang berfungsi untuk memverifikasi *login* yang dilakukan adalah *valid* oleh pengguna yang seharusnya. Pada *form* ini terdapat *textbox* yang harus diisi 6 digit kombinasi angka desimal yang telah sebelumnya dikirimkan oleh firebase melalui SMS kepada nomor telepon yang diisi sebelumnya. *Textbox* akan secara otomatis terisi jika perangkat yang digunakan untuk aplikasi sama dengan perangkat pada nomor telepon terpasang. Jika perangkat berbeda maka *User* harus mengisikan manual kombinasi angka yang dikirimkan melalui SMS ke dalam *textbox* dan menekan tombol *verify* (B) secara manual.

1. Perancangan Tampilan *Form News*

**GAMBAR 3.17**

PERANCANGAN TAMPILAN FORM NEWS ADMIN

**News 4**

**News 3**

**News 2**

**News 1**

**+**

*Sumber: Data Olahan (2018)*

**GAMBAR 3.18**

PERANCANGAN TAMPILAN *FORM* NEWS PARENT

**News 4**

**News 3**

**News 2**

**News 1**

*Sumber: Data Olahan (2018)*

Setelah melakukan *login* maka semua tipe *User* akan dapat melihat postingan berita yang telah di terbitkan oleh *User* admin baris news1… merupakan tampilan berita secara singkat, setiap baris berita dapat ditekan untuk membaca isi berita secara lengkap.

Pada *User* admin dapat melakukan penambahan berita dengan cara menekan tombol “+” sedangkan pada *interface parent* tidak dapat menambahkan berita.

Terdapat tiga buah tombol pada bagian bawah tampilan *form*-*form* utama antara lain:

1. *Button* E : untuk mengarahkan ke *form* *news*
2. *Button* F : untuk mengarahkan ke *form notifikasi*
3. *Button* G : untuk mengarahkan ke *form* *setting*
4. *Button* : untuk kembali ke form sebelumnya.
5. Perancangan *form* *setting*

**GAMBAR 3.19**

PERANCANGAN TAMPILAN FORM SETTING

**L**

**M**

**K**

**J**

**Setting**

*Sumber: Data Olahan (2018)*

Pada *form* setting *User* *interface* admin terdapat empat *Button* pada bagian utama *form* yang terdiri dari :

1. *Button* J yaitu untuk mengarahkan pada *form* *edit user*.
2. *Button* K yaitu untuk mengarahkan ke *form* *class*.
3. *Button* L yaitu untuk mengarahkan ke *form* *info* profil sekolah
4. *Button* M yaitu untuk keluar dari aplikasi.

Sedangkan untuk tiga *Button* di bawah sama dengan penjelasan pada *form* sebelumnya.

Pada *user* *interface* *user* parent tampilan *form* setting berbeda dengan admin karena fungsi-fungsi yang tidak dibutuhkan oleh *user* parent maka pada *user* *interface* parent sebagai berikut :

**GAMBAR 3.20**

PERANCANGAN TAMPILAN FORM SETTING

**L**

**M**

**Setting**

*Sumber: Data Olahan (2018)*

*Button* yang terdapat pada *form* setting ini memiliki fungsi yang sama dengan tapilan *user* *interface* admin.

1. Perancangan Tampilan *Form User*

**GAMBAR 3.21**

PERANCANGAN TAMPILAN FORM USER

**Admin1**

**No telepon**

**Admin2**

**No telepon**

**Admin3**

**No telepon**

**Admin**

**+**

*Sumber: Data Olahan (2018)*

Pada saat *form User* dibuka, maka *User* sudah bisa melakukan pengaturan untuk data admin, baik itu tambah baru, *edit,* ataupun hapus. Beberapa fungsi yang dapat dilakukan pada *form User* ini diantaranya :

1. *Button* + : yaitu digunakan untuk melakukan penambahan *user* admin.
2. Pada sertiap data *user* yang telah dibuat dapat ditekan dan akan menampilkan data *user* sehingga data *user* dapat diubah yaitu nomor telepon dan nama *user* dengan tapilan sebagai berikut.

**GAMBAR 3.22**

PERANCANGAN TAMPILAN FORM EDIT USER

Nama *User*

Field C

Field B

Field A

+62

99999999999

**I**

**H**

*Sumber: Data Olahan (2018)*

Pada sub *form* ini terdapat *field* isian dan *Button* sebagai berikut:

1. *Button* H : merupakan *Button* untuk menyimpan data yang telah diubah,
2. *Button* I : merupakan *Button* untuk menghapus semua data *user* yang sedang diubah dari database.
3. *Field* A merupakan *field* kode negara untuk nomor telepon seluler Indonesia, *field* ini tidak dapat diubah oleh pengguna.
4. *Field* B merupakan *field* nomor telepon tanpa kode negara yang harus diisi ketika meregistrasi *user* admin.
5. *Field* C merupakan *field* nama *user* admin.
6. Perancangan Tampilan *Form notification*

**GAMBAR 3.22**

PERANCANGAN TAMPILAN *FORM* NOTIFICATION

Notifikasi 2

dd-MM-yyyy HH-mm-ss

Notifikasi 1

dd-MM-yyyy HH-mm-ss

**NOTIF**

*Sumber: Data Olahan (2018)*

Setiap aktifitas pemberitahuan yang dilakukan oleh semua tipe *user* aktivitas tersebut akan tercatat pada *database* secara *realtime* kemudian aktivitas tersebut akan tampil di *form* *notification*. Pada *form* ini setiap *record* yang ditampilkan dapat dipilih dan akan mengarah ke isi berita yang terkait.

*Button* pada bagian bawah memiliki sama dengan *form* sebelumnya.

1. Perancangan Tampilan *Form comment*

**GAMBAR 3.23**

PERANCANGAN TAMPILAN FORM COMMENT

**P**

Field comment

Comment 1

Comment 2

**o**

**NEWS**

Isi berita

*Sumber: Data Olahan (2018)*

Setelah sebuah berita dibuat maka semua *user* dapat membuat kometar pada *form* komentardengan memasukkan isi komentar pada *field* komentar. Semua *user* juga dapat melihat komentar yang telah dikirim. Pada *form* ini terdapat *Button* sebagai berikut:

1. *Button* O berfungsi untuk menghapus isi berita yang terpilih *Button* ini hanya terdapat pada *user* *interface* admin.
2. *Button* P berfungsi untuk mengirimkan isi komentar yang telah dimasukkan oleh *User* *Button* ini terdapat pada semua *user* *interface*.