DESAIN

Pada desain terdapat beberapa tampilan dan detail tampilan yang dibuat secara menarik di aplikasi ABC Asyik.

1. DESAIN INTERFACE
2. Tampilan Halaman Menu

Pada tampilan halaman menu terdapat beberapa button yang akan menu ke fitur selanjutnya.



*Gambar 2. Tampilan halaman menu*

1. Tampilan Menu Bermain

Pada tampilan menu bermain terdapat huruf yang terdiri dari huruf besar dan kecil. Jika button pada huruf tersebut ditekan maka akan berbunyi sesuai dengan huruf tersebut (dubbing)



*Gambar 3.Tampilan menu bermain*

1. Tampilan Ayo Belajar

Pada tampilan ayo belajar terdapat evaluasi atau tes untuk user (anak TK). Tes tersebut berupa pertanyaan tentang nama huruf.



*Gambar 4.Tampilan menu Ayo Belajar*

1. Tampilan Peringkat

Pada tampilan peringkat ini terdapat nilai dan akan muncul peringkat yang tertinggi.

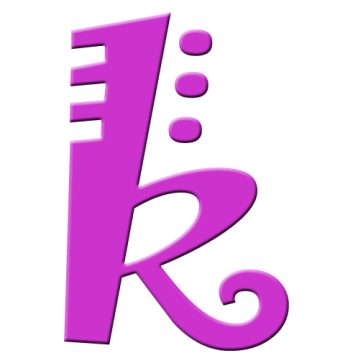


*Gambar 5.Tampilan Peringkat*

1. COMPONENT DESAIN

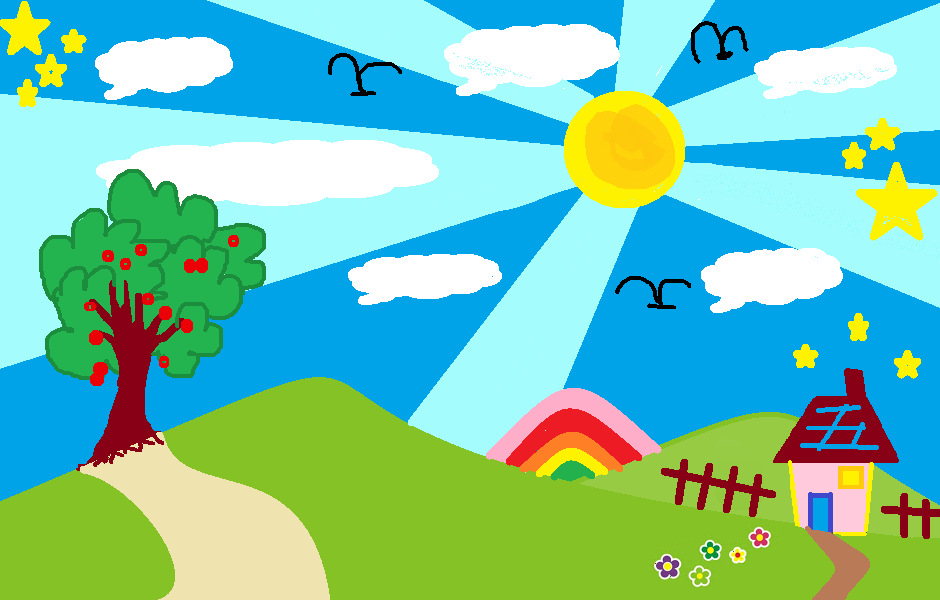
Pada detail tampilan ini terdapat detail tampilan dari aplikasi ABC Asyik.

* Detail Huruf



* Detail Background halaman utama

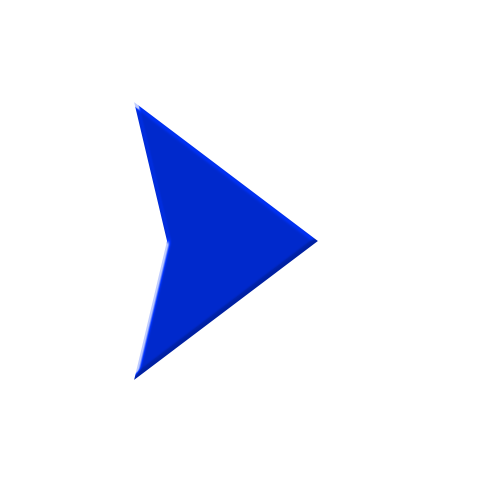
Detail background halaman utama menggambarkan dunia bermain anak TK. Background ini didesain dengan penuh warna yang memberikan kesan menarik dan menggambarkan karakter anak TK.



*Gambar 6.Tampilan background halaman utama*

* Tombol Next

Tombol next diguanakan sebagai buuton untuk menuju ke laman berikutnya.



*Gambar 7.Tampilan tombol Next*

* Tampilan Refresh

Tampilan refresh digunakan sebagai button untuk mengulangi fitur yang sedang diakses oleh user.



*Gambar 8.Tampilan tombol refresh*

* Tampilan Trophy

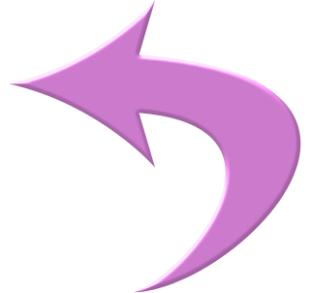
Tampilan trophy ini terdapat pada tampilan peringkat yang merupakan *reward* untuk user yang memperoleh peringkat tertinggi.



*Gambar 9.Tampilan Trophy*

* Tampilan Undo

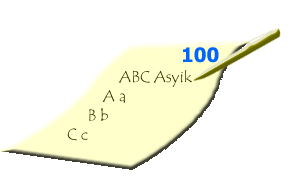
Tampilan undo digunakan sebagai button untuk kembali ke halaman atau perintah sebelumnya.



*Gambar 10.Tampilan Undo*

* Tampilan Ayo Belajar

Tampilan ini



*Gambar 11.Tampilan Ayo Belajar*

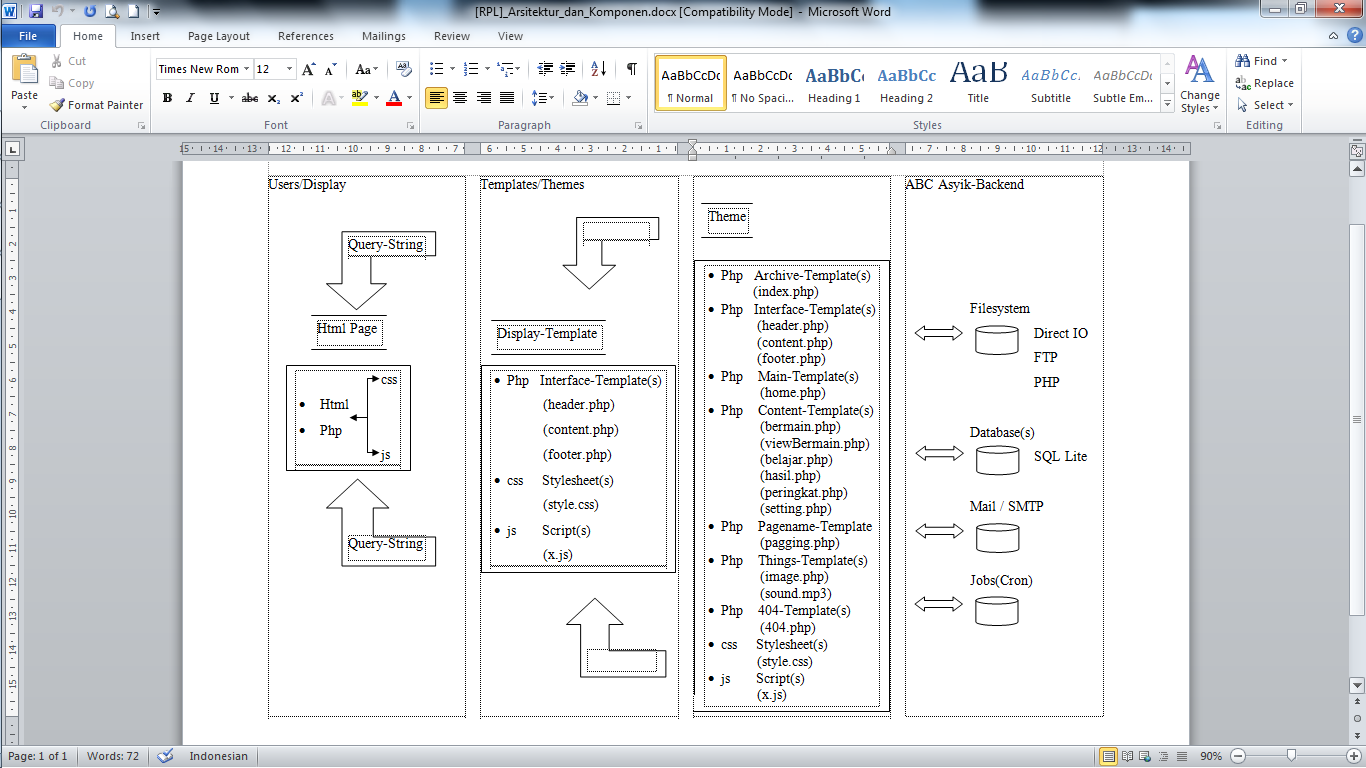
* Tampilan Logo

Tampilan logo merupakan logo aplikasi ABC Asyik yang didesain dengan penuh warna dan menarik.



*Gambar 12.Tampilan Log*

1. ARCHITECTURE DESAIN



1. BEHAVIORAL DIAGRAM

* Detail Activity

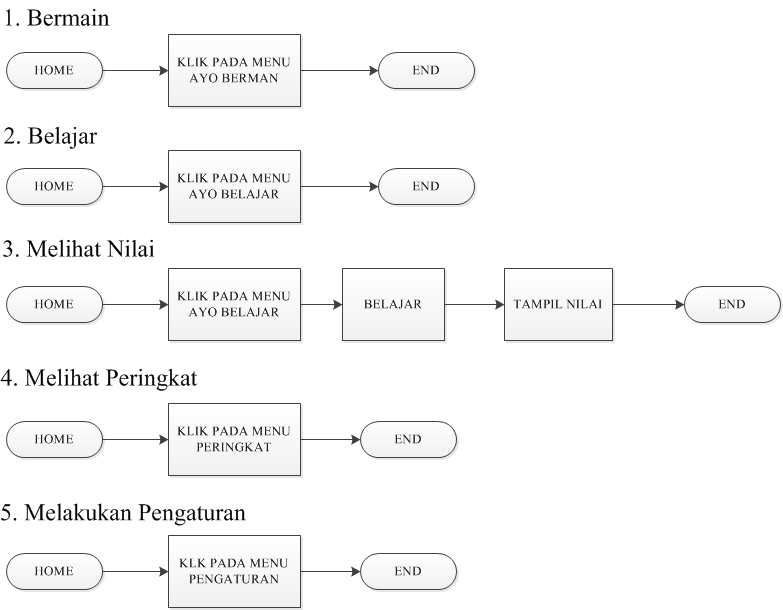
Activity diagrams menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, decision yang mungkin terjadi, dan bagaimana mereka berakhir. Activity diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada

beberapa eksekusi.

Activity diagram merupakan state diagram khusus, di mana sebagian besar state adalahaction dan sebagian besar transisi ditrigger 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-prosesdan 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 vertikal. Activity diagram dapat dibagi menjadi beberapa object *swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu. Pada aplikasi ABC Asyik mempunyai Aktivity Diagram sebagai berikut:



* Use Case Diagram

Use case class digunakan untuk memodelkan dan menyatakan unit fungsi/layanan yang disediakan oleh sistem (bagian sistem: subsistem atau class) ke pemakai. Use case dapat dilingkupi dengan batasan sistem yang diberi label nama sistem. Use case adalah sesuatu yang menyediakan hasil yang dapat diukur ke pemakai atau sistem eksternal.

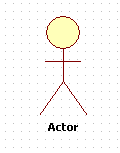
Karakteristik :

* Use cases adalah interaksi atau dialog antara sistem dan actor, termasuk pertukaran pesan dan tindakan yang dilakukan oleh sistem.
* Use cases diprakarsai oleh actor dan mungkin melibatkan peran actor lain. Use cases harus menyediakan nilai minimal kepada satu actor.
* Use cases bisa memiliki perluasan yang mendefinisikan tindakan khusus dalam interaksi atau use case lain mungkin disisipkan.
* Use case class memiliki objek use case yang disebut skenario. Skenario menyatakan urutan pesan dan tindakan tunggal.

Komponen Pembentuk Use Case Diagram :

1. Actor

Pada dasarnya actor bukanlah bagian dari use case diagram, namun untuk dapat terciptanya suatu use case diagram diperlukan beberapa actor. Actor tersebut mempresentasikan seseorang atau sesuatu (seperti perangkat, sistem lain) yang berinteraksi dengan sistem. Sebuah actor mungkin hanya memberikan informasi inputan pada sistem, hanya menerima informasi dari sistem atau keduanya menerima, dan memberi informasi pada sistem. Actor hanya berinteraksi dengan use case, tetapi tidak memiliki kontrol atas use case. Actor digambarkan dengan stick man . Actor dapat digambarkan secara secara umum atau spesifik, dimana untuk membedakannya kita dapat menggunakan *relationship*.

[](http://mitanovia.files.wordpress.com/2009/01/actor.png?w=540)

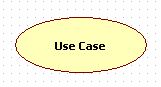
*Gambar 2. Actor*

1. Use Case

Use case adalah gambaran fungsionalitas dari suatu sistem, sehingga customer atau pengguna sistem paham dan mengerti mengenai kegunaan sistem yang akan dibangun.

Cara menentukan Use Case dalam suatu sistem:

* Pola perilaku perangkat lunak aplikasi.
* Gambaran tugas dari sebuah actor.
* Sistem atau “benda” yang memberikan sesuatu yang bernilai kepada actor.
* Apa yang dikerjakan oleh suatu perangkat lunak (\*bukan bagaimana cara mengerjakannya).



*Gambar 3.Use Case*

Relasi dalam Use Case

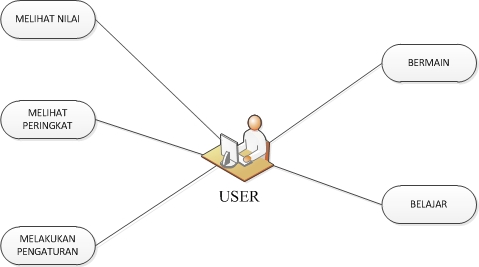
Ada beberapa relasi yang terdapat pada use case diagram:

* Association, menghubungkan link antar element.
* Generalization, disebut juga inheritance (pewarisan), sebuah elemen dapat merupakan spesialisasi dari elemen lainnya.
* Dependency, sebuah element bergantung dalam beberapa cara ke element lainnya.
* Aggregation, bentuk assosiation dimana sebuah elemen berisi elemen lainnya.

Tipe relasi/ stereotype yang mungkin terjadi pada use case diagram:

* <<include>> , yaitu kelakuan yang harus terpenuhi agar sebuah event dapat terjadi, dimana pada kondisi ini sebuah use case adalah bagian dari use case lainnya.
* <<extends>>, kelakuan yang hanya berjalan di bawah kondisi tertentu seperti menggerakkan alarm.
* <<communicates>>, mungkin ditambahkan untuk asosiasi yang menunjukkan asosiasinya adalah communicates association . Ini merupakan pilihan selama asosiasi hanya tipe relationship yang dibolehkan antara actor dan use case.

Pada aplikasi ABC Asyiek Use case diagram yang digunakan adalah:



*Gambar .Use Case Diagram ABC Asyik*