

SISTEM KEAMANAN AKTIVITAS KOMPUTER ANAK BERBASIS OPENSOURCE

Syafri Arlis¹

ABSTRACT

Technological developments at the present time is very rapid, starting from local access to be able to access information of the world. But behind it all the time management in the use of computers should be is arranged, especially in children. Parents must be responsible and can adjust their schedules, especially in scheduling use of a home computer. So that use can be controlled schedule and could make children more independent and disciplined. Designing an application form Parental Control can minimize the child in excessive computer use.

Keywords: *parental control, a computer, activity, schedules*

INTISARI

Perkembangan teknologi pada saat ini sangat pesat, mulai dari akses lokal untuk dapat mengakses informasi dari dunia. Namun di balik itu semua manajemen waktu dalam penggunaan komputer harus diatur, terutama pada anak-anak. Orang tua harus bertanggung jawab dan dapat menyesuaikan jadwal mereka, terutama dalam penggunaan penjadwalan komputer rumah. Sehingga penggunaan dapat dikontrol jadwal dan bisa membuat anak-anak lebih mandiri dan disiplin. Merancang sebuah aplikasi pengawasan komputer dibawah bimbingan orangtua dapat meminimalkan anak dalam menggunakan komputer yang berlebihan.

Kata Kunci: bimbingan orangtua, komputer, aktivitas, jadwal

¹ Dosen Sistem Informasi UPI-YPTK Padang

PENDAHULUAN

Seiring dengan perkembangan dan tingkat kebutuhan, perkembangan teknologi dan informasi di era globalisasi ini semakin meningkat juga. Perkembangan tersebut telah membawa manusia menentukan tingkat efesiensi yang tinggi dalam segala bidang karena kebutuhan akan informasi menjadi hal yang terpenting dalam era globalisasi. Kebutuhan akan informasi di butuhkan dalam semua bidang baik di organisasi, perusahaan, bisnis, instansi dan juga di lembaga pendidikan. Salah satu perubahan yang sangat berpengaruh yaitu pada tingkat aktivitas anak yang tinggi terhadap komputer.

Orangtua tidak luput dari persoalan yang diantaranya bagian pengontrolan anak. Dari persoalan yang ada diharapkan bisa memanfaatkan teknologi informasi yang ada sebagai solusi pemecahan masalah untuk menghasilkan sebuah tujuan yang di harapkan dengan menggunakan sebuah sistem keamanan pada komputer. Dengan adanya sistem keamanan aktivitas komputer, orang tua diharapkan mampu mempermudah dalam proses pengontrolan anak.

Sistem keamanan ini bertujuan untuk menciptakan sebuah pengamanan orang tua kepada anak, karena sudah terlalu banyak hal negatif yang ada pada komputer dan isi-isi nya.

Salah satu aspek metode yang digunakan dalam proses pengamanan adalah sistem pengunci desktop, dengan sistem ini semua aktivitas bisa di blok dan bisa diatur sedemikian rupa.

Dalam kenyataannya masih banyak orang tua yang selalu membiarkan anaknya memakai komputer tanpa ada pengawasan sedikitpun. Melalui sistem ini diharapkan sebuah inovasi baru dalam proses pengamanan aktivitas

anak pada komputer. Berdasarkan latar belakang di atas maka di anggap perlu melakukan penelitian mengenai sistem keamanan aktivitas komputer anak yang cocok diterapkan di keluarga ini sekaligus akan dituliskan dalam bentuk penelitian yang berjudul **“SISTEM KEAMANAN AKTIVITAS KOMPUTER ANAK BERBASIS OPENSOURCE”**

PENDEKATAN PEMECAHAN MASALAH

Konsep Dasar Sistem

1. Pengertian Sistem

Definisi sistem berkembang sesuai dengan konteks dimana pengertian sistem itu digunakan. Berikut akan diberikan beberapa defiisi sistem secara umum:

- a) Kumpulan dari bagian-bagian yang bekerja sama untuk mencapai tujuan yang sama
- b) Sekumpulan objek-objek yang saling berealisasi dan berinteraksi serta hubungan antar objek bisa dilihat sebagai satu kesatuan yang dirancang untuk mencapai satu tujuan

Dengan demikian, secara sederhana sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur atau variabel-variabel yang saling terorganisasi, saling berinteraksi, dan saling bergantung satu sama lain. *Murdick dan Ross (1993)* mendefinisikan sistem sebagai seperangkat elemen yang digabungkan satu dengan lainnya untuk suatu tujuan bersama. Sementara definisi sistem dalam kamus *Webster's Unbridge* adalah elemen yang saling berhubungan dan membentuk suatu kesatuan atau organisasi.

2. Karakteristik Sistem

Suatu sistem mempunyai karakteristik tertentu yakni :

- a. Mempunyai komponen artinya suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi atau bekerja sama membentuk suatu kesatuan, dimana komponen sistem dapat berupa suatu sub sistem.
- b. Mempunyai batas sistem (*boundary*), artinya suatu sistem mempunyai daerah yang membatasi antara satu sistem dengan sistem yang lainnya.
- c. Mempunyai lingkungan luar sistem (*environment*) artinya suatu sistem apa yang mempengaruhi operasi sistem dari sistem itu sendiri. Lingkungan luar ini dapat menguntungkan bagi sistem dan dapat juga merugikan sistem itu sendiri.
- d. Mempunyai penghubung sistem (*interface*) artinya suatu sistem itu merupakan media penghubung antar sub sistem dengan sub sistem lainnya. Keluaran (*output*) suatu sub sistem merupakan masukan (*input*) bagi sub sistem yang lainnya melalui suatu penghubung.
- e. Mempunyai masukan (*input*) artinya suatu sistem masukan merupakan energi yang dimasukkan ke dalam sistem.
- f. Mempunyai pengolah sistem (proses) artinya sistem dapat mempunyai suatu bagian pengolahan yang akan merubah masukan menjadi keluaran.
- g. Mempunyai keluaran (*output*) artinya hasil dari energi yang telah diolah dapat merupakan input bagi sub sistem yang lainnya.
- h. Mempunyai sasaran sistem (*goal*) artinya suatu sistem pasti mempunyai suatu tujuan atau susunan maka

tak ada gunanya operasi dari suatu sistem tersebut jika tidak ada sasaran (*goal*) yang ingin dicapai, sebab sasaran ini merupakan yang menentukan sekali masukan dan hasil yang akan dikeluarkan.

3. Klasifikasi Sistem

Didalam Buku Konsep Sistem Informasi oleh **Tata Sutabri, 2012** suatu sistem merupakan suatu bentuk integrasi antara satu komponen dengan komponen lain karena sistem memiliki sasaran yang berbeda untuk setiap kasus yang terjadi yang ada didalam sistem tersebut. Oleh karena itu, sistem dapat diklasifikasikan dari beberapa sudut pandang, diantaranya:

- a. Sistem abstrak dan sistem fisik
- b. Sistem alamiah dan sistem buatan manusia
- c. Sistem determinasi dan sistem probabilistik
- d. Sistem terbuka dan sistem tertutup

4. Pengertian Sistem Informasi

Sistem informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan (Tata Sutabri 2012:46).

Komponen Sistem Informasi

1. Blok masukan (*input block*)
Input mewakili data yang masuk ke dalam sistem informasi. Input yang dimaksud adalah metode dan media untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen-dokumen dasar.

2. Blok model (*model block*)
Blok ini terdiri dari kombinasi prosedur, logika, dan model matematik yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara yang sudah tertentu untuk menghasilkan keluaran yang diinginkan.
3. Blok keluaran (*output block*)
Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.
2. Blok teknologi (*technology block*)
Teknologi merupakan “*tool box*” dalam sistem informasi. Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran, dan membantu pengendalian dari sistem secara keseluruhan. Teknologi terdiri dari 3 (tiga) bagian utama, yaitu teknisi (*brainware*), perangkat lunak (*software*), dan perangkat keras (*hardware*).
3. Blok basis data (*database block*)
Kumpulan data yang saling berkaitan dan berhubungan antara satu dengan yang lain, tersimpan di perangkat keras komputer dan menggunakan perangkat lunak untuk memanipulasinya. Data perlu disimpan dalam basis data untuk keperluan penyediaan informasi lebih lanjut. Data di dalam basis data perlu diorganisasikan sedemikian rupa supaya informasi yang dihasilkan berkualitas. Organisasi basis data yang baik juga berguna untuk

efisiensi kapasitas penyimpanannya. Basis data diakses atau dimanipulasi menggunakan perangkat lunak paket yang disebut DBMS (*Database Management Sistem*).

4. Blok kendali (*control block*)
Banyak hal yang dapat merusak sistem informasi seperti bencana alam, api, temperatur, air, debu, kecurangan-kecurangan, kegagalan dari sistem itu sendiri, ketidakefisienan, sabotase, dan lain sebagainya. Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah ataupun bila terlanjur terjadi kesalahan-kesalahan dapat langsung cepat diatasi (Sumber: Tata Sutabri, 2012:47-48).

Siklus Hidup Pengembangan Sistem

1. Pengertian SDLC

SDLC atau (*Software Development Life Cycle*) atau sering disebut juga *Sistem Development Life Cycle* adalah proses mengembangkan atau merubah suatu sistem perangkat lunak dengan menggunakan model-model dan metedologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya (berdasarkan *best practice* atau cara-cara yang sudah teruji baik).

Tahapan-tahapan yang ada pada SDLC secara global adalah sebagai berikut:

- Inisiasi (*initiation*)
Tahap ini biasanya ditandai dengan pembuatan proposal proyek perangkat lunak.

- Pengembangan konsep sistem (*sistem concept development*)
Mendefinisikan lingkup konsep termasuk dokumen lingkup sistem, analisis manfaat biaya, manajemen rencana, dan pembelajaran kemudahan sistem.
- Perencanaan (*planning*)
Mengembangkan rencana manajemen proyek dan dokumen perencanaan lainnya. Menyediakan dasar untuk mendapatkan sumber daya (*resources*) yang dibutuhkan untuk memperoleh solusi.
- Analisis kebutuhan (*requirements analysis*)
Menganalisis kebutuhan pemakai sistem perangkat lunak (*user*). membuat dokumen kebutuhan fungsional.
- Desain (*design*)
Mentransformasikan kebutuhan detail menjadi kebutuhan yang sudah lengkap, dokumen desain sistem fokus pada bagaimana dapat memenuhi fungsi-fungsi yang dibutuhkan.
- Pengembangan (*development*)
Mengonversi desain ke sistem informasi yang lengkap termasuk bagaimana memperoleh dan melakukan instalasi lingkungan sistem yang dibutuhkan.
- Integrasi dan pengujian (*integration and test*)
Mendemonstrasikan sistem perangkat lunak bahwa telah memenuhi kebutuhan yang dispesifikasikan pada dokumen kebutuhan fungsional. Dengan diarahkan oleh staf penjamin kualitas (*quality assurance*) dan *user*. Menghasilkan laporan analisis pengujian.
- Implementasi (*implementation*)
Termasuk pada persiapan implementasi, implementasi perangkat lunak pada lingkungan produksi (lingkungan pada *user*) dan menjalankan resolusi dari permasalahan yang teridentifikasi dari fase integrasi dan pengujian.
- Operasi dan pemeliharaan (*operation dan maintenance*)
Mendeskrripsikan pekerjaan untuk mengoperasikan dan memelihara sistem informasi pada lingkungan produksi. Termasuk pada implementasi akhir dan masuk pada proses peninjauan.
- Disposisi (*dispotition*)
Mendeskrripsikan aktifitas akhir dari pengembangan sistem dan membangun data yang sebenarnya sesuai dengan aktifitas *user*.

Pengertian Data dan Pengolahan Data

Mengenai pengertian data, John J.Longkutoy dalam bukunya pengenalan komputer akan mendefinisikan sebagai berikut. "istilah data adalah suatu istilah majemuk yang berarti fakta atau bagian dari fakta yang mengandung arti yang dihubungkan dengan kenyataan, simbol-simbol, gambar-gambar, angka-angka, huruf-huruf, atau symbol-simbol yang menunjukkan suatu ide, objek, kondisi, atau situasi dan lain-lain. Jelasnya data itu bisa berupa apa saja dan dapat ditemui dimana saja. Kegunaan data adalah sebagai bahan dasar yang objektif (relatif) didalam proses kebijaksanaan dan keputusan oleh pimpinan organisasi" (Sumber:Tata Sutabri,2012:2).

Alat Bantu Dalam Menganalisa Sistem

1. Unified Modeling language
Unified Modelling Language (UML) adalah sebuah “bahasa” yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem.

Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan class operation dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa-bahasa berorientasi objek seperti c++, Java, C# atau VB.NET. Meskipun demikian UML juga dapat digunakan untuk modeling aplikasi procedural dalam VB atau C. Berikut ini adalah penjelasan mengenai berbagai diagram UML serta tujuannya:

2. Usecase diagram

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

3. Class Diagram

Diagram kelas mendeskripsikan jenis-jenis objek dalam sistem dan berbagai hubungan statis yang terdapat diantara mereka. Diagram kelas juga menunjukkan property dan operasi sebuah kelas dan batasan-batasan yang terdapat dalam hubungan-hubungan objek tersebut.

4. Activity Diagram

Activity diagram menggambarkan *workflow* (aliran kerja) atau aktifitas dari sebuah

sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwasanya diagram aktifitas menggambarkan aktifitas sistem bukan apa yang dilakukan actor, jadi aktivitas yang dapat dilakukan oleh system.

5. Sequence Diagram

Sequence Diagram menggambarkan kelakuan / perilaku objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram sekuense maka harus diketahui objek-objek yang terlibat dalam sebuah usecase beserta metode-metode yang dimiliki kelas.

Alat Bantu Perancangan Logika Program

Untuk keperluan membaca logika program digunakan alat bantu perancangan program yang menjelaskan urutan-urutan logis dari program yang dibuat. Disini penulis menggunakan Bagan Alir Program (Program Flowchart)

Sistem Parental Control

Konsep mengatur jadwal aktivitas komputer anak, disebut parental control system dari kata parental(*keluarga*) dan control(*pengatur*). Parental Control sendiri mempunyai pengertian yaitu Fitur baru yang dirancang untuk membebaskan orang tua memberikan kontrol untuk penggunaan komputer anak-anak mereka. Parental Controls biasanya dapat dijalankan dalam bentuk aplikasi atau sudah termasuk didalam browser.

1. Parental Control

Sistem kendali atau sistem kontrol (control system) adalah suatu alat (kumpulan alat) untuk mengendalikan, memerintah, dan mengatur keadaan dari suatu sistem. Istilah sistem kendali ini dapat

dipraktekkan secara manual untuk mengendalikan stir mobil pada saat kita mengendarai/menyetir mobil kita, misalnya, dengan menggunakan prinsip loloh balik. Dalam sistem yang otomatis, alat semacam ini sering dipakai untuk peluru kendali sehingga peluru akan mencapai sasaran yang diinginkan. Banyak contoh lain dalam bidang industri / instrumentasi dan dalam kehidupan kita sehari-hari di mana sistem ini dipakai. Alat pendingin (AC) merupakan contoh yang banyak kita jumpai yang menggunakan prinsip sistem kendali, karena suhu ruangan dapat dikendalikan sehingga ruangan berada pada suhu yang kita inginkan..

Konsep Dasar Object Pascal

1. Sekilas Mengenai Pascal

Object Pascal merupakan turunan dari bahasa pascal yang dikembangkan oleh tim dari Apple Computer dengan konsultan Niklaus Wirth sebagai penemu Pascal. Object Pascal diperlukan untuk mendukung MacApp, yakni sebuah pengembangan dari framework aplikasi Macintosh. Namun pengembangan oleh Apple dihentikan, karena Apple lebih memilih mengembangkan chip Motorola 68K untuk IBM PowerPC yang ditulis dalam bahasa C++.

Pada Tahun 1986, Borland juga mengenal Object Pascal dalam produk yang dinamakan Turbo Pascal dalam produk yang dinamakan Turbo Pascal sudah tersedia untuk DOS dan baru tersedia untuk sistem operasi Windows pada tahun 1994 dengan nama Delphi.

Perkembangan selanjutnya, Object Pascal banyak dikembangkan dengan berbagai macam IDE, compiler(*cross compiler*), ekstension, dan interpreter. Berikut ini adalah beberapa versi pengembangan

Object Pascal dengan dialek yang berbeda namun tetap menggunakan sintaks dan semantik Pascal (Handayani Saptaji W, 2012:11).

- a. Borland Pascal, yang dikemudian diumumkan secara komersial dengan nama '*Delphi*'.
- b. Embarcadero Technologies, yang membeli Delphi pada tahun 2008. Pada versi Delphi XE sudah mampu mengkompilasi sintaks Pascal untuk Windows dan Mac OS X.
- c. .NET mendukung Delphi 8 sampai dengan Delphi 2007, kemudian namanya berganti menjadi '*Oxygene*' namun tidak penuh kompatibel dengan dialek Pascal.
- d. Morfik Pascal yang dikembangkan oleh Morfik dengan target AJAX untuk aplikasi web. Compiler Morfik mampu mengkompilasi object Pascal menjadi Kode yang berjalan di browser dan server.
- e. Free Pascal, sebuah proyek opensource yang mampu menghasilkan aplikasi yang mampu berjalan di banyak platform sistem operasi, seperti Linux(32/64 bit), FreeBSD, Mac OS, Solaris, Win32, Win64, Windows CE, iOS dan Android.
- f. MIDlet Pascal, yang dapat menghasilkan kode biner java dari object Pascal.
- g. GNU Pascal, yang mengimplementasikan standar Pascal sesuai ISO 7185 dan standar '*Extended Pascal*' untuk ISO 10206
- h. Turbo51, yakni IDE untuk menghasilkan kode biner yang berjalan di IC mikrokontroler keluarga intel 8051 dari object pascal.
- i. WDSibyl, sebuah IDE opensource seperti Delphi

yang tersedia untuk Microsoft.

2. Free Pascal Compiler

Compiler yang bertugas untuk mengubah kode pascal yang merupakan bahasa pemrogram level tinggi (*high level programming*) yang dimengerti manusia menjadi kode biner (*low level programming*) yang dimengerti prosessor sebagai otak komputer. Free Pascal Compiler (FPC) merupakan salah satu compiler yang bersifat open source yang dikembangkan oleh komunitas www.freepascal.org. FPC didesain untuk mendukung berbagai macam arsitektur prosessor seperti inter x86, x86-64 bit, Power PC, SPARC dan ARM dan bermacam sistem operasi seperti windows, linux, freeBSD, Mac OS/Mac OS X, Palm OS, mendukung kompatibilitas dengan bahasa assemblre yang banyak digunakan untuk perangkat keras yang lebih spesifik. Karena memiliki banyak kompatibilitas itulah komunitas freePascal mengkampanye slogan '*write once, compile anywhere*' (Handayani Saptaji W, 2012:13).

3. Lazarus IDE di CodeTyphon (Sekarang Typhon IDE)

CodeTyphon sendiri merupakan produk freeware dari pilotlogic yang menawarkan instalasi Lazarus yang sekaran disebut Typhon IDE dan komponen-komponennya dalam satu paket instalasi terintegrasi. Paket instalasi Code Typhon juga sudah dilengkapi dengan berbagai macam compiler cross platform (FPC) yang dapat menghasilkan kode biner yang dapat dieksekusi oleh banyak tipe prosessor. Dan hebatnya lagi, sebuah paket Code Typhon bisa diinstal di bermacam-macam sistem operasi. (Handayani Saptaji W, 2012:53).

IDE lazarus yang sekarang disebut Typhon ini memiliki tampilan dan tool pendukung seperti Delphi. Dalam satu lingkungan IDE lazarus tersedia jendela Menu, jendela *Object Inspector* untuk mengatur properti dan even tiap komponen, jendela source editor untuk menuliskan kode pascal, jendela *Form Designer* untuk mendesain tampilan grafis aplikasi dan jendela *Message* untuk mengetahui proses kompilasi. (Handayani Saptaji W, 2012:17).

Analisa Sistem

Analisa sistem dapat didefinisikan sebagai penguraian dari suatu sistem informasi yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya.

Analisa masalah dapat dilakukan agar penemuan masalah yang ada, dapat diketahui penyebabnya, sehingga nantinya dari analisa masalah tersebut didapatkan suatu bentuk pemecahan masalah.

1. Analisa Sistem yang Sedang Berjalan

Analisa sistem yang berjalan pada OS windows sebelumnya hanya terbatas pada akses blokir dan belum bisa menjadwalkan kapan sistem ini bisa diakses atau.

Alternatif Pemecahan Masalah

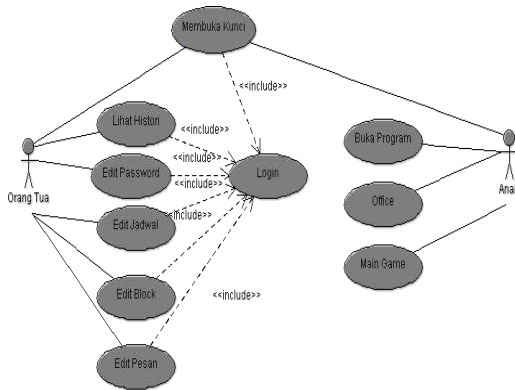
Dari permasalahan yang didapat, maka penulis mencoba untuk membangun suatu aplikasi yang dapat membantu dan memudahkan dalam proses mengamankan komputer, dimana aplikasi ini dapat menjadwalkan

kanan komputer bisa diakses atau tidak bisa diakses.

Dengan dibangunnya aplikasi ini, dapat kita lihat juga dibawah pembahasan mengenai pemodelan dengan menggunakan Unified Modelling Language (UML) yang berguna untuk mempermudah proses rancangan. Tidak lupa juga design perancangan tampilan dari setiap menu.

1. Diagram Use Case Sistem

Diagram *Use case* sistem itu menggambarkan bagaimana aktifitas dari masing-masing aktor didalam sistem tersebut.



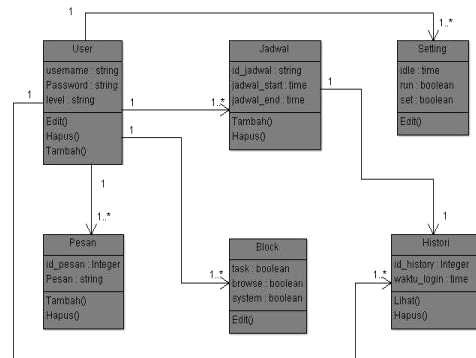
Gambar 1. Use Case Diagram Sistem

2. Class Diagram

Class diagram menggambarkan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem. Kelas-kelas menggambarkan sesuatu yang ditangani dalam sistem. Hubungan antar kelas dalam bentuk asosiasi (terhubung antar kelas), dependensi (satu kelas tergantung atau menggunakan kelas lainnya), spesialisasi (satu kelas adalah sebuah spesialisasi dari kelas lainnya) atau pemaketan (tergabung menjadi suatu unit).

Sebuah *class diagram* menggambarkan semua hubungan bersama dengan struktur dari kelas

yaitu atribut dan operasi. Diagram dikatakan statis artinya bahwa struktur dideskripsikan sebagai valid dari pada semua posisi di dalam siklus kerja sistem.

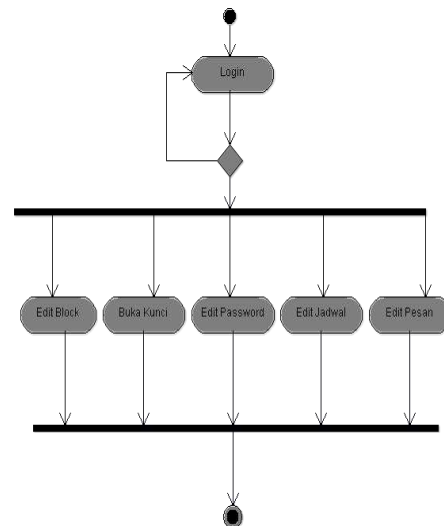


Gambar 2. Class Diagram

3. Activity Diagram

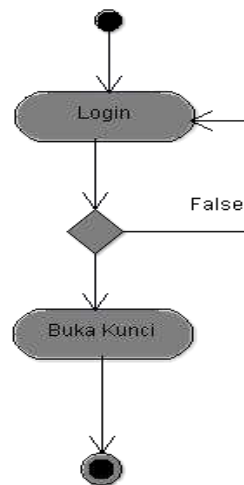
Activity diagram merupakan diagram yang menjelaskan kegiatan masing-masing aktor dalam sistem tersebut. Kegiatan masing-masing aktor tersebut dapat dilihat pada ulasan berikut ini.

a. Activity Admin/Orang Tua



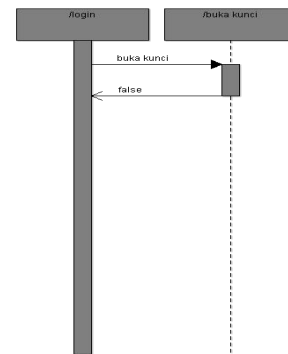
Gambar 3. Activity Admin

b. Activity User/Anak



Gambar 4. Activity Diagram Anak

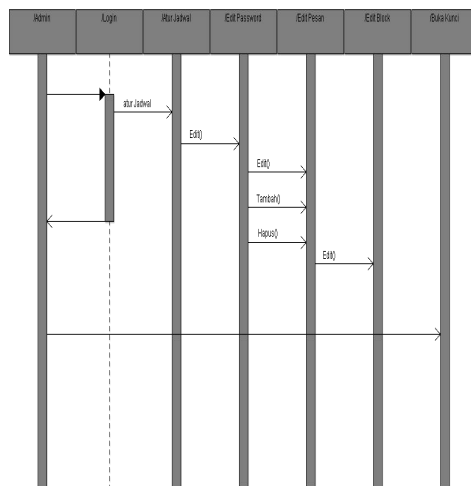
b. Sequence Diagram Anak



Gambar 6. Sequence Diagram Anak

4. **Sequence Diagram**
Sequence diagram menggambarkan kegiatan objek pada use case di dalam sistem. Kegiatan setiap objek tersebut dijelaskan pada gambar-gambar berikut ini.

a. Admin



Gambar 5. Sequence Diagram Admin

HASIL DAN PEMBAHASAN Implementasi Sistem

Tahap implementasi system merupakan salah satu tahap dalam daur hidup pengembangan sistem, dimana tahap ini merupakan tahap meletakkan system informasi supaya siap untuk dipakai. Dalam tahap ini berlangsung beberapa aktifitas secara berurutan yakni mulai dari menerapkan rencana implementasi, melakukan kegiatan implementasi, dan tindak lanjut implementasi.

Pengujian Sistem

Pengujian sistem menggambarkan bagaimana sebuah system berjalan. Tahapan pengujian ini berisikan hasil eksekusi program dan penjelasan program yang dibuat untuk mendukung sistem yang telah dirancang.

Adapun aplikasi yang dibutuhkan dalam pengujian system ini yaitu

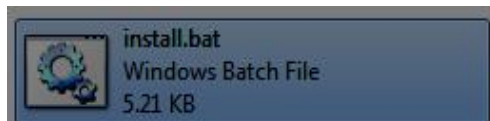
1. Sistem Operasi Microsoft Window 7 Home Premium
2. Bahasa Pemrograman Typhone64
3. Notepad++

Proses Instalasi

1. Instalasi Typhone

Pada pengujian sistem yang diterapkan pada program ini dibutuhkan sebuah *IDE*, maka dijalankan terlebih dahulu menginstall aplikasi *CodeTyphon*. Langkah-langkah dalam instalasi software *CodeTyphon* sebagai berikut :

- a. Klik dua kali pada icon Instal bat

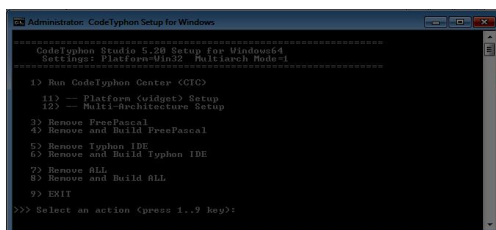


Gambar 7. Typhon64.exe

- b. Muncul jendela seperti gambar dibawah ini, pilih 0 untuk instalasi

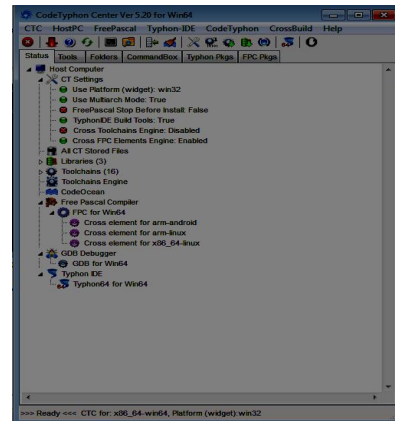
Gambar 8. CodeTyphon64

- c. Lalu akan muncul setup baru, dapat dilihat pada gambar 9.



Gambar 9. Menu Setup.bat

- d. Selanjutnya tekan 8 dan tunggu proses instalasi sampai selesai dan muncul seperti dibawah ini



Gambar 10. Menu Codetyphon

- e. Maka klik dua kali typhone64 for win64



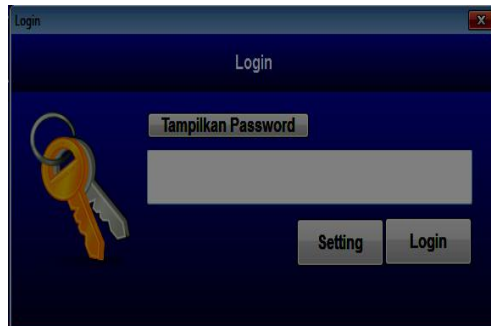
Gambar 11. Splash Screen Typhon

Hasil Pengujian Sistem

Pengujian sistem akan menggambarkan bagaimana sebuah sistem berjalan. Tahapan pengujian ini berisikan hasil eksekusi program dan penjelasan program yang telah dibuat untuk mendukung sistem yang telah dirancang.

1. Form Login

Form ini hanya ditujukan untuk user/admin login. Di halaman ini juga tempat proses login setting, dapat dilihat pada gambar 12 dibawah ini :



Gambar 12. Form Login

2. Form Setelan Admin untuk jadwal

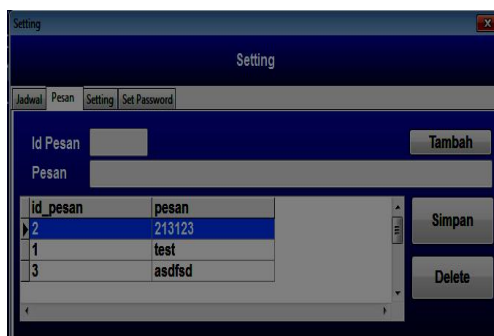
Halaman ini admin dapat mengedit semuanya, mulai dari jadwal dan lain-lain . Dapat dilihat pada gambar 13 berikut :



Gambar 13. Form Setelan jadwal

3. Form setelan admin untuk Pesan

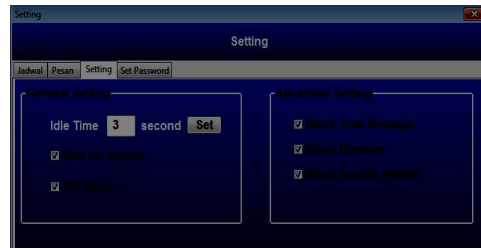
Form ini digunakan untuk proses pengaturan pesan yang tampil, seperti gambar 14 dibawah ini :



Gambar 14. Form Pengaturan Pesan

4. Form Setelan umum

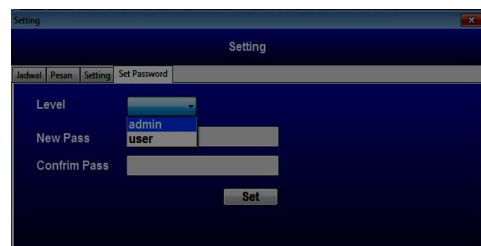
Halaman ini digunakan oleh admin sebagai tempat pengaturan sistem yang akan di hentikan atau dilarang, seperti gambar 15 dibawah ini :



Gambar 15. Form Pengaturan

5. Form Setelan Admin untuk mengatur Password

Form ini berguna untuk mengatur password admin dan user itu sendiri, seperti pada gambar 16 dibawah ini :



Gambar 16. Form Setelan Password

KESIMPULAN

Berdasarkan hasil analisa dan desain yang telah dilakukan dan dibahas pada bab-bab sebelumnya, maka dapat disimpulkan bahwa :

1. Dengan dirancangnya sistem keamanan ini, pihak orang tua dapat mengatur aktivitas komputer anaknya.
2. Dengan adanya sistem keamanan ini dapat mengatur jadwal komputer terkunci atau tidak dan bisa membatasi anak untuk melakukan aktivitas komputer.
3. Keterbatasan kemampuan sistem pada aplikasi ini yaitu:

- a. Aplikasi masih terbatas untuk mengakses sistem OS utama yakni Windows OS.
- b. Masih terbatas pada pengaturan jadwal dan kunci aplikasi umum di windows.
- c. Hanya untuk sistem operasi Windows.

DAFTAR PUSTAKA

- [1] Al Fatta, Hanif. 2007. *Analisis Dan Perancangan Sistem Informasi*. Yogyakarta. Andi Offset.
- [2] Jogiyanto, H.M. 2005. *Analisis dan Desain Sistem Informasi*. Yogyakarta: Andi Offset.
- [3] Rosa, Salahudin. *Modul Pembelajaran Rekayasa Perangkat Lunak (terstruktur dan berorientasi objek)*, Bandung: modula.
- [4] Munawar. *Pemodelan Visual dengan UML*. Yogyakarta: Graha Ilmu
- [5] Nazir, Mohammad. 1998. *Metode Penelitian*. Jakarta. Ghalia Indonesia.
- [6] Oetomo, Dharma. Sutedja, Budi. 2006. *Perancangan dan Pembangunan Sistem Informasi*. Yogyakarta: Andi Offset.
- [7] Saptaji W, Handayani. 2012. *Bikin Android dengan Pascal? BISA!*. Surabaya: Widya Media.
- [8] Sugiarti, Yuni. 2013. *Analisis dan Perancangan UML (Unified Modelling Language)*. Yogyakarta: Graha Ilmu
- [9] Sutabri, Tata. 2012. *Pengenalan Sistem Informasi*. Jakarta: Elex Media Komputindo.