

BAB II

Landasan Teori

A. E-Voting

Menurut (Syam, 2019) E-Voting adalah suatu sistem pemilihan dimana data dicatat disimpan dan diproses dalam bentuk informasi digital. Karakteristik sistem e-voting mencakup tiga variabel yaitu kerahasiaan, kemudahan, dan kegunaan, yang kemudian dianalisis pengaruhnya terhadap keinginan pemilih (Khusnia, 2018). Syarat e-voting atau pemilu melalui elektronik antara lain pemilu tetap berdasarkan asas langsung, umum, bebas rahasia (Luber) dan (Jurdil) jujur adil (Grahitandaru, 2014). Dengan memanfaatkan teknologi komputerisasi dengan menggunakan aplikasi pemilihan ketua OSIS (e-voting) menjadikan kegiatan pemilihan ketua OSIS lebih mudah dalam penyampaian informasi, lebih cepat dalam melakukan perhitungan suara, serta membuat lebih hemat terhadap kertas sehingga hasil yang didapat bisa efektif dan efisien. Fungsi lain voting dapat digunakan untuk mengumpulkan aspirasi siswa dan sebagai jalan keluar untuk menyelesaikan suatu masalah. Alasan penggunaan sistem informasi e-voting ini yaitu dapat mengatasi hal-hal atau masalah yang sering terjadi pada sistem pemilihan berbasis konvensional terkait proses perhitungan yang lambat, biaya yang cukup besar dan kurang efektif. Dengan adanya e-voting ini nantinya dapat mengatasi masalah tersebut dan dapat memberikan kemudahan kepada pemilih dalam melakukan pemilihan terkait ketua OSIS SMP Islam Al-A'rof Cilacap.

B. QR Code (*Quick Response Code*)

Menurut (Widayati, 2007) Kode QR atau biasa dikenal dengan QR Code adalah bentuk evolusi kode batang dari satu dimensi menjadi dua dimensi. Kode QR fungsionalitas utama yaitu dapat dengan mudah dibaca oleh pemindai QR (*quick response*) atau respon cepat, yang sesuai dengan tujuannya adalah untuk menyampaikan informasi dengan cepat dan mendapatkan respons yang cepat pula. Pengguna hanya mengaktifkan program pembaca kode QR, mengarahkan kamera ke kode QR, selanjutnya program pembaca kode QR akan secara otomatis meminde data yang telah tertanam pada kode QR. Jika kode QR berisi alamat suatu situs, maka pengguna dapat langsung mengakses situs tersebut tanpa harus lebih dulu mengetikkan alamat dari situs yang dituju. Hal ini tentu mempermudah pengguna dalam mendapatkan informasi yang ditawarkan.

C. Basis Data

Menurut (Amborowati, 2007) Basis Data adalah susunan atau kumpulan data operasional lengkap dari suatu organisasi atau perusahaan yang diorganisir atau dikelola dan disimpan secara terintegrasi dengan menggunakan metode tertentu menggunakan computer sehingga mampu menyediakan informasi optimal yang diperlukan pemakainya.

Menurut (Amborawati, 2007) Sistem Basis Data adalah suatu sistem menyusun dan mengelola record-record menggunakan komputer untuk menyimpan atau merekam serta memelihara data operasional lengkap sebuah

organisasi atau perusahaan sehingga mampu menyediakan informasi yang optimal yang diperlukan pemakai untuk proses pengambilan keputusan.

Komponen-komponen pokok sistem basis data menurut (Amborowati, 2007) yaitu:

1) Data

Data didalam sebuah basis data dapat disimpan secara terintegrasi dan data dapat dipakai secara bersama-sama (*shared*). Data disimpan secara terintegrasi yaitu basis data yaitu merupakan kumpulan dari berbagai macam *file* dari aplikasi yang berbeda yang disusun dengan cara menghilangkan bagian-bagian yang rangkap (*redundant*). Data dipakai bersama-sama atau *shared* yaitu masing-masing bagian dari basis data dapat diakses oleh pemakai dalam waktu yang bersamaan untuk aplikasi yang berbeda. Jenis data yang hubungannya pada basis data menurut (Amborowati, 2007) yaitu:

- a. Data operasional, merupakan data dari satu organisasi berupa data yang disimpan didalam basis data.
- b. Data masukan (*input data*), merupakan data dari luar sistem yang dimasukkan melalui peralatan input yang dapat mengubah data operasional.
- c. Data keluaran (*output data*), merupakan data berupa laporan melalui peralatan output sebagai hasil proses dari dalam suatu sistem yang mengakses data operasional.

2) *Hardware* (perangkat keras)

Terdiri dari semua peralatan komputer yang digunakan untuk pengelolaan sistem basis data, yang berupa peralatan untuk menyimpan basis data yaitu *secondary storage* (disk, drum dan lain-lain), peralatan *input* dan *output*, peralatan komunikasi data dan lain-lain.

3) *Software* (perangkat lunak)

Berfungsi sebagai perantara (*interface*) antara pemakai dengan data fisik pada basis data. *Software* pada basis data dapat berupa:

- a. *DBMS (Database Management Sistem)* yang menangani akses terhadap basis data sehingga pemakai tidak perlu memikirkan proses penyimpanan dan pengelolaan data secara *detail*.
- b. Program-program aplikasi dan *procedure-procedure*.

4) *User* (pemakai)

Pemakai basis data dibagi atas tiga klasifikasi menurut Amborowati (2007:3) yaitu:

- a. *Database Administrator (DBA)*, merupakan orang atau tim yang bertugas mengelola sistem basis data secara keseluruhan.
- b. *Programmer*, merupakan orang atau tim yang bertugas membuat program aplikasi, misalnya untuk perbankan, administrasi, akuntansi, dan lain-lain.
- c. *End User*, orang yang mengakses basis data melalui terminal dengan menggunakan *query language* atau program aplikasi yang dibuat oleh programmer.

D. *System Development Life Cycle (SDLC)*

SDLC atau *Software Development Life Cycle* atau sering disebut juga *System Development Life Cycle* adalah proses pengembangan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya berdasarkan *best practice* atau cara-cara yang sudah teruji baik (Shalahuddin, 2013).

Tahapan-tahapan yang ada pada SDLC secara global menurut (Shalahuddin, 2013) sebagai berikut:

a) Inisiasi (*initiation*)

Tahap ini biasanya ditandai dengan pembuatan proposal proyek perangkat lunak.

b) Pengembangan konsep sistem (*system concept development*)

Mendefinisikan lengkap konsep termasuk dokumen lingkup sistem, analisis manfaat biaya, manajemen rencana, dan pembelajaran kemudahan sistem.

c) Perancangan (*planning*)

Pengembangan rencana manajemen proyek dan dokumen perencanaan lainnya. Menyediakan dasar untuk mendapatkan sumber daya (*resources*) yang dibutuhkan untuk memperoleh solusi.

d) Analisis kebutuhan (*requirements analysis*)

Menganalisis kebutuhan pemakai sistem perangkat lunak (*user*) dan mengembangkan kebutuhan *user*.

e) Desain (*design*)

Mentransformasikan kebutuhan detail menjadi kebutuhan yang sudah lengkap, dokumen desain sistem fokus pada bagaimana dapat memenuhi fungsi-fungsi yang dibutuhkan.

f) Pengembangan (*development*)

Mengonversi desain ke sistem informasi yang lengkap termasuk bagaimana memperoleh dan melakukan instalasi lingkungan sistem yang dibutuhkan.

g) Integrasi dan pengujian (*integrasi and test*)

Mendemokrasikan sistem perangkat lunak bahwa telah memenuhi kebutuhan yang dispesifikasikan pada dokumen kebutuhan fungsional.

h) 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.

i) Operasi dan pemeliharaan (*operations and maintenance*)

Mendeskripsikan pekerjaan untuk mengoperasikan dan memelihara sistem informasi pada lingkungan produksi (lingkungan pada *user*), termasuk implementasi akhir dan masuk pada proses peninjauan.

j) Disposisi (*disposition*)

Mendesripsikan aktifitas akhir dari pengembangan sistem dan membangun data yang sebenarnya sesuai dengan aktifitas *user*.

E. Prototype

Menurut (Jogiyanto, 2005) ”*Prototype* adalah bentuk dasar atau model awal dari suatu sistem atau bagian dari suatu sistem”.

Model *prototype* biasanya menyederhanakan permasalahan yang ada sehingga tetap ada kemungkinan terjadi perbedaan antara kinerja model *prototype* yang sudah dibuat dengan sistem operasional yang sesungguhnya.

Langkah-langkah pengembangan dengan metode *prototype* menurut (Nugroho, 2008) diantaranya:

a) Identifikasi kebutuhan organisasi

Tahap ini seperti tahap perancangan dan analisa dengan waktu dan kegiatan yang didapatkan, dengan sumber wawancara diambil secara sempel saja.

b) Pengembangan *Prototype*

Tahap pengembangan dapat memakai perangkat lunak yang cocok untuk pengembangan cepat, misalnya perangkat lunak jenis *DBMS (Database Management System)*

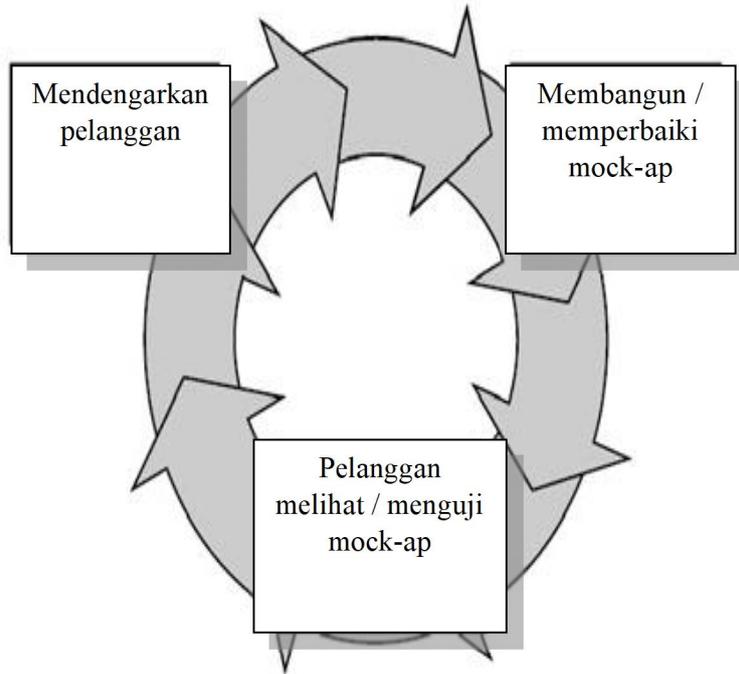
c) Uji *Prototype* dan Evaluasi

Tahap ini bertujuan untuk mengetahui apakah *prototype* dapat diterima. Apabila tidak diterima maka *prototype* harus disempurnakan lagi untuk nantinya diuji kembali dan apabila sudah diterima maka *prototype* dapat digunakan.

d) Penggunaan *Prototype*

Pada tahap ini *prototype* digunakan statusnya sudah bukan *prototype* lagi, melainkan sistem yang operasional.

Berikut adalah gambar dari model *prototype*:



Gambar 2.1 Ilustrasi Pendekatan *Prototype*

Keterangan:

Mack-up adalah sesuatu yang digunakan sebagai model desain yang digunakan untuk mengajar, demonstrasi, evaluasi desain, promosi, atau keperluan lain (Shalahuddin, 2013). Sebuah *mock-up* disebut sebagai *prototype* perangkat lunak jika menyediakan atau mampu mendemonstrasikan sebagian besar fungsi sistem perangkat lunak dan memungkinkan pengujian desain sistem perangkat lunak. Iterasi terjadi

pada pembuatan *prototype* sampai sesuai dengan keinginan pelanggan (*costomer*) atau *user*.

F. Unified Modelling Language (UML)

Definisi UML menurut (Muhammada, 2015), "Teknik pemrograman yang berorientasi objek dan merupakan bahasa pemodelan untuk pembangunan perangkat lunak." UML merupakan salah satu standar metodologi pengembangan sistem informasi dan atau perangkat lunak berorientasi objek. Salah satu *software* yang paling banyak digunakan untuk melakukan *design software* melalui pendekatan UML adalah *Rational Rose*. *Rational Rose* adalah kaskas (*tools*) pemodelan *visual* untuk pengembangan sistem berbasis objek yang sangat handal untuk digunakan sebagai bantuan bagi para pengembang dalam melakukan analisis dan perancangan sistem. Pada *Rasional Rose* ada beberapa diagram diantaranya *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, dan *Class Diagram*.