

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang

Dalam dunia pendidikan, teknologi informasi sangat banyak membantu seperti dalam hal proses pengolahan data, baik itu data siswa, guru, administrasi sekolah maupun data akademik dalam rangka meningkatkan kualitas pendidikan sekolah tersebut. TK ST. Maria Assumpta adalah salah satu TK yang mulai menerapkan teknologi informasi dalam proses pengolahan data akademik.

Di TK ST. Maria Assumpta memiliki 120 siswa, yang terbagi menjadi 3 kelas yaitu kelas A dengan jumlah 44 siswa dan kelas B yang terbagi lagi menjadi 2 kelas yang berjumlah 76 siswa. Dengan jumlah guru 10 orang dan pegawai 3 orang yang terdiri dari guru PNS 5 orang dan non PNS 8 orang.

TK ST. Maria Assumpta dalam pengolahan data telah menggunakan aplikasi Microsoft Excel tanpa *database* yang masih sangat sederhana. Dengan menggunakan aplikasi tersebut masih terdapat beberapa kekurangan. Pada saat pengolahan data dapat terjadi duplikasi dalam peng-*input*-an data karena tidak ada informasi yang menunjukkan peringatan penambahan data tersebut.

Pada saat melakukan perubahan data yang sudah tersimpan pegawai TK ST. Maria Assumpta harus mencari satu per satu data tersebut sampai data ditemukan, sehingga membutuhkan waktu karena banyaknya data. Semua itu dikarenakan belum adanya media penyimpanan data yang teratur dalam bentuk *database*. Dengan kondisi sistem ini TK ST Maria Assumpta mengalami kesulitan dalam hal

pencarian data dan pembuatan laporan serta belum adanya SDM yang terampil dalam bidang teknologi informasi yang dapat mengoptimalkan penggunaan perangkat komputer yang ada pada TK ST. Maria Assumpta.

Beberapa kendala di atas dapat ditangani dengan dikembangkannya sistem informasi akademik. Diharapkan dengan sebuah sistem informasi akademik pegawai tata usaha dapat terbantu dalam pengolahan data, penyimpanan data dan pembuatan laporan secara efisien dan efektif.

Menyadari perlunya untuk meningkatkan kinerja dari bagian tata usaha sekolah ini, maka perlu dioptimalkan pula penggunaan sumber daya yang ada melalui otomatisasi sistem informasi akademik sekolah. Berdasarkan latar belakang tersebut maka dilakukan penelitian dengan judul **“SISTEM INFORMASI AKADEMIK TK ST. MARIA ASSUMPTA KUPANG”**.

## **1.2. Rumusan Masalah**

Berdasarkan dari latar belakang yang telah dikemukakan maka rumusan masalah adalah bagaimana membangun sistem informasi akademik pada TK Sta. Maria Assumpta Kupang yang dapat terbantu dalam pengolahan data, penyimpanan data dan pembuatan laporan secara efisien dan efektif?

## **1.3. Batasan Masalah**

Batasan masalah dalam penelitian ini yaitu sebagai berikut:

1. Sistem ini akan menghasilkan keluaran berupa laporan absen, form penilaian, program semester dan rencana kegiatan mingguan.

2. Bahasa pemrograman yang akan dipakai adalah bahasa pemrograman *Java Netbeans* dengan *Database MySQL*.

## **1.4. Tujuan dan Manfaat Penelitian**

### **1.4.1. Tujuan Penelitian**

Tujuan dari penelitian ini adalah merancang bangun sebuah Sistem Informasi Akademik TK Sta Maria Assumpta Kupang untuk meningkatkan efektifitas dan efisiensi kerja dari bagian tata usaha sekolah.

### **1.4.2. Manfaat penelitian**

Adapun beberapa manfaat yang akan diperoleh dalam pembuatan sistem ini adalah:

1. Bagi TK Sta. Maria Assumpta Kupang

Dengan adanya sistem informasi akademik yang akan dibuat ini dapat membantu mempermudah kerja dari bagian tata usaha dalam mengolah data-data akademik secara cepat dan akurat.

2. Bagi Peneliti

Dengan adanya sistem informasi akademik yang akan dibuat ini, dapat menjadi acuan dan wawasan pengetahuan dalam merancang sebuah sistem informasi akademik.

## **1.5. Metode Penelitian**

Menurut Pressman (2010), model *waterfall* adalah model klasik yang bersifat statis, berurutan dalam membangun *software*. Nama model ini sebenarnya adalah *Linear Sequential Model*. Model ini sering disebut dengan *classic life cycle*

atau model *waterfall*. Model ini pertama kali yang diperkenalkan oleh Winston Royce sekitar tahun 1970 sehingga sering dianggap kuno, tetapi merupakan model yang paling banyak dipakai didalam *Software Engineering* (SE). Model ini melakukan pendekatan secara sistematis dan berurutan. Disebut dengan *waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan.

*Waterfall* mengusulkan sebuah pendekatan kepada pengembangan *software* yang sistematis dan sekuensial yang mulai dari tingkat kemajuan sistem pada seluruh analisis, desain, kode, pengujian dan pemeliharaan. Model ini melingkupi aktivitas-aktivitas sebagai berikut : rekayasa dan pemodelan sistem informasi, analisis kebutuhan, desain, koding, mengujian dan pemeliharaan.

Model pengembangan ini bersifat linear dari tahap awal pengembangan system yaitu tahap perencanaan sampai tahap akhir pengembangan system yaitu tahap pemeliharaan. Tahapan berikutnya tidak akan dilaksanakan sebelum tahapan sebelumnya selesai dilaksanakan dan tidak bisa kembali atau mengulang ke tahap sebelumnya.

Tahapan atau fase model *waterfall* ini adalah gambar tahapan atau fase yang paling umum tentang model *waterfall*. Akan tetapi Roger S. Pressman memecah model ini menjadi 6 tahapan meskipun secara garis besar sama dengan tahapan-tahapan model *waterfall* pada umumnya. Berikut adalah Gambar dan penjelasan dari tahap-tahap yang dilakukan di dalam model ini menurut Pressman:

1. ***System Information Engineering and Modeling.***

Permodelan ini diawali dengan mencari kebutuhan dari keseluruhan sistem yang akan diaplikasikan ke dalam bentuk *software*. Hal ini sangat penting, mengingat *software* harus dapat berinteraksi dengan elemen-elemen yang lain seperti *hardware*, *database*. Tahap ini sering disebut dengan *Project Definition*.

2. ***Software Requirements Analysis.***

Proses pencarian kebutuhan diintensifkan dan difokuskan pada *software*. Untuk mengetahui sifat dari program yang akan dibuat, maka para *software engineer* harus mengerti tentang domain informasi dari *software*, misalnya fungsi yang dibutuhkan, *user interface*. Dari 2 aktivitas tersebut (pencarian kebutuhan sistem dan *software*) harus didokumentasikan dan ditunjukkan kepada pelanggan.

3. ***Design.***

Proses ini digunakan untuk mengubah kebutuhan-kebutuhan diatas menjadi representasi ke dalam bentuk *blueprint software* sebelum *coding* dimulai. Desain harus dapat mengimplementasikan kebutuhan yang telah disebutkan pada tahap sebelumnya. Seperti 2 aktivitas sebelumnya, maka proses ini juga harus didokumentasikan sebagai konfigurasi dari *software*.

4. ***Coding.***

Untuk dapat dimengerti oleh mesin, dalam hal ini adalah komputer, maka desain tadi harus diubah bentuknya menjadi bentuk yang dapat dimengerti oleh mesin, yaitu ke dalam bahasa pemrograman melalui proses *coding*. Tahap ini merupakan implementasi dari tahap *design* yang secara teknis nantinya dikerjakan oleh *programmer*.

## 5. **Testing / Verification.**

Sesuatu yang dibuat haruslah diujicobakan. Demikian juga dengan *software*. Semua fungsi-fungsi *software* harus diujicobakan, agar *software* bebas dari *error*, dan hasilnya harus benar-benar sesuai dengan kebutuhan yang sudah didefinisikan sebelumnya.

## 6. **Maintenance.**

Pemeliharaan suatu *software* diperlukan, termasuk di dalamnya adalah pengembangan, karena *software* yang dibuat tidak selamanya hanya seperti itu. Ketika dijalankan mungkin saja masih ada *error* kecil yang tidak ditemukan sebelumnya, atau ada penambahan fitur-fitur yang belum ada pada *software* tersebut. Pengembangan diperlukan ketika adanya perubahan dari eksternal perusahaan seperti ketika ada pergantian sistem operasi, atau perangkat lainnya.

Dalam model ini terdapat beberapa sifat-sifat yang menonjol dan cenderung menjadi permasalahan pada model *waterfall*.

1. Ketika *problem* muncul, maka proses berhenti karena tidak dapat menuju ke tahapan selanjutnya. Apabila terdapat kemungkinan *problem* tersebut muncul akibat kesalahan dari tahapan sebelumnya, maka proses harus membenahi tahapan sebelumnya agar *problem* ini tidak muncul.
2. Karena pendekatannya secara *sequential*, maka setiap tahap harus menunggu hasil dari tahap sebelumnya. Hal itu tentu membuang waktu yang cukup lama, artinya bagian lain tidak dapat mengerjakan hal lain selain hanya menunggu hasil dari tahap sebelumnya.

Meskipun seringkali kebutuhan sistem tidak dapat didefinisikan secara eksplisit yang diinginkan, tetapi paling tidak, *problem* pada kebutuhan sistem di awal *project* lebih ekonomis dalam hal uang (lebih murah), usaha, dan waktu yang terbuang lebih sedikit jika dibandingkan *problem* yang muncul pada tahap-tahap selanjutnya.

Meskipun demikian, karena model ini melakukan pendekatan secara urut / *sequential*, maka ketika suatu tahap terhambat, tahap selanjutnya tidak dapat dikerjakan dengan baik dan itu menjadi salah satu kekurangan dari model ini. Kapan model *waterfall* di gunakan? Salah satu model tradisional dan mudah yang tahapannya mengalir satu arah seperti air terjun adalah *Waterfall Model* atau *Linear Sequential Model*. Pertanyaannya, kapan sebaiknya model tersebut digunakan?

Teori-teori lama menyimpulkan ada beberapa hal, yaitu:

1. Ketika semua persyaratan sudah dipahami dengan baik di awal pengembangan.
2. Definisi produk stabil dan tidak ada perubahan saat pengembangan untuk alasan apapun seperti perubahan eksternal, perubahan tujuan, perubahan anggaran atau perubahan teknologi. Untuk itu, teknologi yang digunakan pun harus sudah dipahami dengan baik.
3. Menghasilkan produk baru, atau versi baru dari produk yang sudah ada. Sebenarnya, jika menghasilkan versi baru maka sudah masuk incremental development, yang setiap tahapnya sama dengan *Waterfall* kemudian diulang-ulang.

4. *Porting* produk yang sudah ada ke dalam *platform* baru. Dengan demikian, *Waterfall* dianggap pendekatan yang lebih cocok digunakan untuk proyek pembuatan sistem baru. Tetapi salah satu kelemahan paling dasar adalah menyamakan pengembangan perangkat keras dengan perangkat lunak dengan meniadakan perubahan saat pengembangan. Padahal, galat diketahui saat perangkat lunak dijalankan, dan perubahan-perubahan akan sering terjadi.

Tahap – tahap pengembangan *waterfall* model adalah :

1. Analisis dan definisi persyaratan Pelayanan, batasan, dan tujuan sistem ditentukan melalui konsultasi dengan *user*.
2. Perancangan sistem dan perangkat lunak Kegiatan ini menentukan arsitektur sistem secara keseluruhan.
3. Implementasi dan pengujian unit Perancangan perangkat lunak direalisasikan sebagai serangkaian program.
4. Integrasi dan pengujian sistem Unit program diintegrasikan atau diuji sebagai sistem yang lengkap untuk menjamin bahwa persyaratan sistem telah terpenuhi
5. Operasi dan pemeliharaan Merupakan fase siklus yang paling lama. Sistem diinstall dan dipakai. Perbaikan mencakup koreksi dari berbagai *error*, perbaikan dan implementasi unit sistem dan pelayanan sistem.

Keuntungan dari Model *Waterfall*, yaitu:

1. Merupakan model pengembangan paling handal dan paling lama digunakan.
2. Cocok untuk system *software* berskala besar.



3. Cocok untuk *system software* yang bersifat generik.
4. Pengerjaan *project system* akan terjadwal dengan baik dan mudah dikontrol.

Kelemahan *Waterfall*, yaitu:

1. Waktu pengembangan lama. hal ini dikarenakan input tahap berikutnya adalah *output* dari tahap sebelumnya. Jika satu tahap waktunya molor, maka waktu keseluruhan pengembangan juga ikut molor.
2. Biaya juga mahal, hal ini juga dikarenakan waktu pengembangan yang lama
3. Terkadang perangkat lunak yang dihasilkan tidak akan digunakan karena sudah tidak sesuai dengan *requirement* bisnis *customer*. hal ini juga dikarenakan waktu pengembangan yang lama. selain itu dikarenakan *waterfall* merupakan aliran yang *linear*, sehingga jika *requirement* berubah proses tidak dapat diulang lagi.

Karena tahap-tahapan pada *waterfall* tidak dapat berulang, maka model ini tidak cocok untuk pemodelan pengembangan sebuah proyek yang memiliki kompleksitas tinggi.

Meskipun *waterfall* memiliki banyak kelemahan yang dinilai cukup fatal, namun model ini merupakan dasar bagi model-model lain yang dikembangkan setelahnya.

## **1.6. Sistematika Penulisan**

Sistematika penyusunan laporan tugas akhir ini merupakan gambaran umum tentang seluruh isi laporan yang terdiri atas 6 (enam) bab, sebagai berikut :

**a. BAB I PENDAHULUAN**

Bab ini berisi uraian tentang latar belakang penulisan, batasan masalah, perumusan masalah, tujuan penelitian, metode penelitian dan sistematika penulisan.

**b. BAB II LANDASAN TEORI**

Bab ini akan dibahas mengenai teori-teori yang mendukung dalam penulisan skripsi ini. Diantaranya penjelasan mengenai aplikasi, internet, website, *Hypertext Transfer Protokol* (HTTP), *Hypertext Markup Language* (HTML), PHP (Hypertext Preprocessor), MySQL, Macromedia Dreamweaver 8, dan Microsoft Visio 2007.

**c. BAB III ANALISIS DAN DESAIN SISTEM**

Bab ini berisi tentang analisis sistem dan perancangan sistem berdasarkan permasalahan yang ditemukan.

**d. BAB IV IMPLEMENTASI SISTEM**

Bab ini menjelaskan proses tindak lanjut dari hasil perancangan sistem sebelumnya.

**e. BAB V ANALISIS HASIL**

Bab ini memuat analisis hasil program dalam hal ini mengenai eksekusi program serta kelebihan dan kekurangan dari program yang dibuat.

## **f. BAB VI PENUTUP**

Dalam bab ini memuat kesimpulan dan saran-saran dari proses perancangan.