

BAB II

LANDASAN TEORI

2.1 Studi Kepustakaan

Penulis menggunakan acuan yang bersumber pada penelitian-penelitian sebelumnya, baik dalam bentuk skripsi, teori maupun jurnal yang dapat dijadikan sebagai pembandingan serta bahan referensi dalam menyelesaikan tugas akhir ini.

Amelia Agustina (2016) dalam penelitiannya mengembangkan sistem informasi *monitoring* persediaan stok kayu di PT Musi Hutan Persada menggunakan metode *Framework Analytical System Thinking* (FAST). Dalam melakukan pencatatan data produksi dan persediaan pada masing-masing cabang PT. Musi Hutan Persada dilakukan secara manual dengan pengolahan data menggunakan *excel*, sehingga memerlukan waktu yang cukup lama untuk melakukan pencarian dan pengolahan data persediaan dan produksi yang dibutuhkan oleh para *manager*. Informasi tersebut tidak dapat dilihat secara langsung oleh manajer pusat. Penelitian ini dapat memonitor persediaan stok kayu pada masing-masing cabang. Sistem dibangun dengan berbasis *online* sehingga manajer pusat dapat memonitor proses produksi dan persediaan dengan menggunakan metode *Framework Analytical System Thinking* (FAST) yakni metode menganalisa, mengorganisir dan mencatat fungsi-fungsi dari suatu sistem, produk, rancangan, proses, prosedur, fasilitas suplai untuk menstimulasi pemikiran dan kreatifitas menggunakan diagram teknik yang memperlihatkan secara grafik fungsi-fungsi dari sebuah *item*, sistem atau prosedur

Sistem ini memiliki persamaan yaitu keduanya sama-sama sistem *monitoring* pada sebuah perusahaan kayu. Dengan sistem, manajer atau atasan dapat mengetahui ketersediaan kayu dan laporan dari bawahan. Keduanya juga dibangun dengan sistem berbasis *online*, sehingga atasan bisa melihat ketersediaan kayu kapan pun dan dimana pun. Namun, keduanya juga memiliki perbedaan yang signifikan. Sistem yang akan penulis bangun lebih kompleks dibandingkan dengan sistem sebelumnya. Sistem ini memonitor kayu sejak berada di TPn, TPK, hingga dermaga. Masing-masing memiliki laporan yang berbeda dan harus dilaporkan kepada atasan disetiap divisinya. Sehingga dipastikan akan berbeda penerapannya. Selain itu, sistem ini juga akan diterapkan dengan berbasis *mobile*, sehingga ringan digunakan dibandingkan dengan berbasis *web*.

Iham Daut (2014) dalam penelitiannya menjelaskan bahwa proses *monitoring* material gudang PT. PLN masih dilakukan secara manual. Proses yang meliputi material masuk, material keluar dan stok masih dilakukan menggunakan *Microsoft Excel*. Setelah semua proses transaksi selesai, barulah bagian gudang melaporkan semua transaksi kepada atasan. Atasan juga baru bisa melihat ketersediaan bahan material setelah karyawan melaporkan hasil kerja tersebut. Sistem *monitoring* yang dibangun ini menggunakan bahasa pemrograman Delphi 7 dan *database Microsoft Access 2007*.

Penelitian ini memiliki persamaan dengan penelitian yang akan penulis bangun. Keduanya sama-sama merupakan sistem *monitoring*. Dengan sistem ini, atasan bisa memonitoring ketersediaan bahan-bahan baik kayu ataupun material PLN. Dengan sistem ini juga, bawahan bisa dengan mudah dalam membuat

laporan kepada atasan. Namun keduanya memiliki perbedaan, penelitian terdahulu diterapkan pada sistem *monitoring* pada PT.PLN sedangkan penelitian yang akan penulis bangun akan diterapkan untuk memonitoring ketersediaan kayu dan laporan keluar masuknya kayu. Dengan begitu, penerapannya juga pasti akan berbeda satu sama lain. Selain itu, sistem yang akan penulis bangun berbasis *mobile*, sedangkan pada sistem sebelumnya dibangun berbasis dekstop. Bahasa pemrograman juga berbeda, dimana penulis menggunakan bahasa pemrograman menggunakan PHP dan *database* MySQL sedangkan pada penelitian sebelumnya menggunakan bahasa pemrograman Delphi dan *database* menggunakan *Microsoft Access*.

Dodi Wahyugi (2008) dalam penelitiannya menjelaskan jika pengembangan sistem pelaporan dan *monitoring* Badan Perencanaan Pembangunan Daerah (BAPPEDA) Kabupaten Bengkulu Utara masih dilakukan secara manual. Pada sistem lama yang berjalan, Satuan Kerja Perangkat Daerah (SKPD) dari seluruh Kabupaten Bengkulu Utara harus melalui tahapan yang cukup lama untuk melaporkan realisasi program mereka kepada BAPPEDA. Padahal, setiap laporan diharapkan akan meningkatkan kualitas serta pengendalian *monitoring* dan evaluasi, transparansi, dan membantu pembuat kebijakan dalam mengambil langkah-langkah kebijakan terhadap pelaksanaan pembangunan di Kabupaten Bengkulu Utara. Sistem ini dibangun berbasis *web*, dengan bahasa pemrograman PHP dan *database* MySQL.

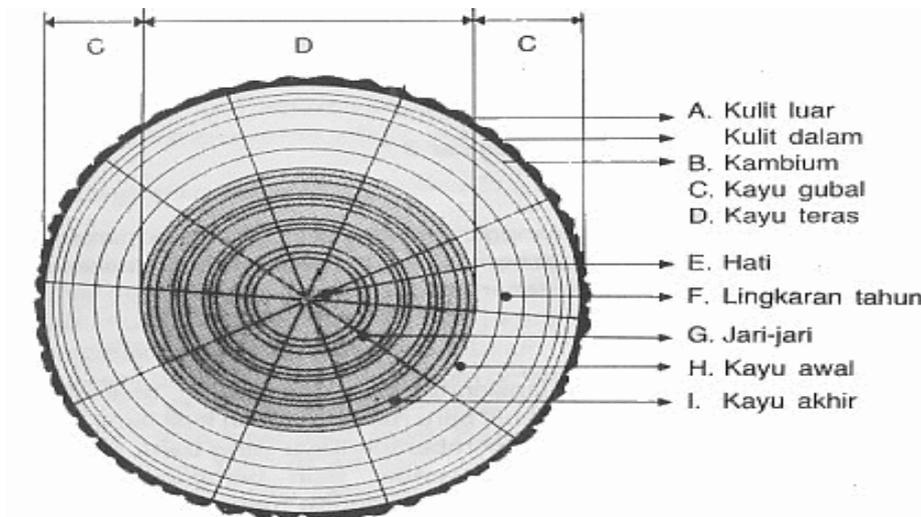
Persamaan sistem ini dengan sistem yang akan penulis bangun adalah sama-sama merupakan sistem *monitoring* dan pelaporan. Bahasa yang pemograman

yang juga sama, yakni menggunakan PHP, sementara *database*-nya juga sama-sama menggunakan MySQL. Namun keduanya memiliki perbedaan yang sangat signifikan, terutama dalam penerapannya. Pada penelitian terdahulu, diterapkan pada *monitoring* BAPPEDA, sedangkan yang akan penulis bangun, diterapkan pada *monitoring* ketersediaan kayu. Selain itu, sistem di atas dibangun dengan berbasis *web*. Sementara sistem yang penulis buat sudah berbasis *mobile* yang lebih mudah dan lebih ringan.

2.2 Dasar Teori

2.2.1 Kayu

Kayu merupakan hasil hutan dari sumber kekayaan alam, merupakan bahan mentah yang mudah diproses untuk dijadikan barang sesuai kebutuhan. Pengertian kayu disini adalah sesuatu bahan yang diperoleh dari hasil pemungutan pohon-pohon di hutan, yang merupakan bagian dari pohon tersebut, setelah diperhitungkan bagian-bagian mana yang lebih banyak dapat dimanfaatkan untuk sesuatu tujuan penggunaan dalam bentuk kayu pertukangan, kayu industri maupun kayu bakar. Bagian-bagian kayu ditampilkan pada gambar 2.1 dibawah ini.



Gambar 2.1 Bagian-Bagian Kayu

1. Kulit

Kulit terdapat pada bagian terluar. Ada dua bagian yaitu :

- a. Kulit bagian luar yang mati, mempunyai ketebalan yang bervariasi menurut jenis pohon.
- b. Kulit bagian dalam yang bersifat hidup dan tipis.

Kulit berfungsi sebagai pelindung bagian-bagian yang terdalam, terhadap kemungkinan pengaruh dari luar yang bersifat merusak, misalnya iklim, serangan serangga, hama, kebakaran serta perusak-perusak kayu lainnya. Selain itu berfungsi sebagai jalan bahan makanan dari daun ke bagian-bagian tanaman.

2. Kambium

Kambium merupakan jaringan yang lapisannya tipis dan bening. Pertumbuhan kambium mlingkari kayu, ke arah luar membentuk kulit baru menggantikan kulit lama yang telah rusak dan ke arah dalam

membentuk kayu yang baru. Pertumbuhan kambium ke arah luar mengakibatkan pohon lambat laun bertambah besar.

3. Kayu gubal

Bagian kayu yang masih muda terdiri dari sel-sel yang masih hidup, terletak di sebelah dalam kambium dan berfungsi sebagai penyalur cairan dan tempat penimbunan zat-zat makanan. Tebal lapisan kayu gubal bervariasi menurut jenis pohon. Pohon yang tumbuh cepat mempunyai lapisan kayu gubal lebih tebal dibandingkan dengan kayu terasnya. Kayu gubal biasanya mempunyai warna terang.

4. Kayu Teras

Kayu teras terdiri dari sel-sel yang dibentuk melalui perubahan-perubahan sel hidup pada lingkaran kayu gubal bagian dalam, disebabkan terhentinya fungsi sebagai penyalur cairan dan lain-lain proses kehidupan. Ruang dalam kayu teras dapat mengandung berbagai macam zat yang member warna lebih gelap. Pohon jenis tertentu kayu teras banyak mengandung bahan-bahan ekstraktif yang memberi keawetan dan membuat lebih berat pada kayu, tetapi tidak semua jenis kayu yang memiliki zat ekstraktif sudah dapat dipastikan keawetannya.

5. Hati

Hati merupakan bagian kayu yang terletak pada pusat lingkaran tahun (tidak mutlak pada pusat bontos). Hati berasal dari kayu awal, yaitu bagian kayu yang pertama kali dibentuk oleh kambium. Hati mempunyai sifat rapuh atau sifat lunak.

6. Lingkaran tahun

Lingkaran tahun tumbuh antara kayu yang terbentuk pada permulaan dan pada akhir suatu musim. Lingkaran-lingkaran tahun ini menunjukkan umur pohon, apabila pertumbuhan diameter membesar terganggu oleh musim kering karena penguuguran daun, ataupun serangan serangga/hama, maka lingkaran tahun dapat terdiri lebih dari satu lingkaran tahun (lingkaran tumbuh) dalam satu musim yang sama. Lingkaran tahun dapat mudah dilihat pada beberapa jenis kayu daun lebar. Pada pohon jenis-jenis tertentu, lingkaran tahun ada kalanya sulit dibedakan terutama di daerah tropik, karena pertumbuhan praktis berlangsung sepanjang tahun.

7. Jari-jari

Jari-jari dari luar ke dalam berpusat pada sumbu batang, berfungsi sebagai tempat saluran bahan makanan yang mudah diproses di daun guna pertumbuhan pohon

2.2.2 Sistem

Sistem adalah sekelompok elemen-elemen yang terintegrasi dengan tujuan yang sama untuk mencapai tujuan. Sistem juga dapat diartikan suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, terkumpul bersama-sama untuk melakukan suatu kegiatan atau tujuan tertentu. Suatu sistem memiliki karakteristik atau sifat-sifat tertentu, yaitu : komponen sistem, batas sistem,

lingkungan luar sistem, penghubung sistem, masukan sistem, keluaran sistem, pengolah sistem, dan sasaran system (Jogiyanto, 2005).

2.2.3 Monitoring

Monitoring didefinisikan sebagai siklus kegiatan yang mencakup pengumpulan, peninjauan ulang, pelaporan, dan tindakan atas informasi suatu proses yang sedang diimplementasikan. Umumnya, monitoring digunakan dalam checking antara kinerja dan target yang telah ditentukan. Pelaku *monitoring* merupakan pihak-pihak yang berkepentingan dalam proses, baik pelaku proses (*self monitoring*) maupun atasan atau *supervisor* pekerja. Berbagai macam alat bantu yang digunakan dalam pelaksanaan sistem *monitoring*, baik observasi atau *interview* secara langsung, dokumentasi maupun aplikasi visual.

Pada dasarnya, *monitoring* memiliki dua fungsi dasar yang berhubungan, yaitu *compliance monitoring* dan *performance monitoring*. *Compliance monitoring* berfungsi untuk memastikan proses sesuai dengan harapan atau rencana. Sedangkan, *performance monitoring* berfungsi untuk mengetahui perkembangan organisasi dalam pencapaian target yang diharapkan. *Output monitoring* berupa progress report proses. *Output* tersebut diukur secara deskriptif maupun non-deskriptif. *Output monitoring* bertujuan untuk mengetahui kesesuaian proses telah berjalan. *Output monitoring* berguna pada perbaikan mekanisme proses atau kegiatan di mana *monitoring* dilakukan (Mercy, 2005).

2.2.4 Sistem Informasi

Sistem informasi merupakan suatu komponen yang mencakup sejumlah komponen (manusia, teknologi informasi, dan prosedur kerja), ada sesuatu yang diproses (data menjadi informasi), dan dimaksudkan untuk mencapai suatu sasaran atau tujuan. Sistem informasi juga dapat diartikan sebagai kombinasi antara prosedur kerja, informasi, orang dan teknologi informasi yang diorganisasikan untuk mencapai tujuan dalam sebuah organisasi (Kadir, 2014).

2.2.5 Aplikasi *Mobile*

Aplikasi *mobile* dikenal dalam dua bentuk dasar. Bentuk pertama adalah aplikasi natif, merupakan aplikasi yang diprogram dan berjalan di perangkat. Bentuk lain adalah aplikasi *web mobile* yang berjalan di dalam *browser web* pada perangkat *mobile*. Aplikasi natif punya banyak manfaat antara lain cepat, punya akses pada semua potensi *platform* di mana aplikasi dibangun. Bagaimanapun, aplikasi natif punya satu keterbatasan, yaitu *portable*. Apabila anda menginginkan aplikasi yang tersedia secara lintas *platform*, paling tidak Anda harus menulis aplikasinya dalam berbagai bahasa pemrograman yang menghasilkan banyak basis kode yang akan merepotkan dalam pengelolaannya, atau harus menggunakan *platform* untuk mendukung laporan abstraksi seperti Titanium atau PhoneGap.

Disisi lain, aplikasi *web mobile* dibuat dengan HTML, CSS, dan JavaScript. Aplikasi tersebut seharusnya dapat berjalan di *browser web* yang dimiliki oleh perangkat *mobile*. Hal ini berarti bukan hanya satu basis kode yang perlu dikelola, tapi juga masih perlu mempertimbangkan variasi *browser web* di setiap platform.

Pengembangan *multiplatform* cukup menjadi tantangan dilingkungan pengembangan aplikasi natif karena setiap *platform* membutuhkan SDK yang berbeda-beda. Kerena perhatian yang sesungguhnya ada pada kemudahan pengembangan dan penerapan pada banyak perangkat *mobile*, pengembangan *web mobile* menjadi solusi yang sangat tepat untuk kebanyakan situasi. Istilah aplikasi *web* (*webapp*) punya banyak konsep yang mirip dan bersinonim, seperti aplikasi *web mobile*, *widget*, *hybrid*, aplikasi HTML5, dan masih banyak lagi.

Secara khusus, aplikasi *mobile web* berbeda dari situs *web* kebanyakan dalam hal tujuannya. Sebuah aplikasi *web* punya perhatian lebih pada *user interface* yang meniru aplikasi *mobile* natif. Ini tetap dibuat dengan teknologi *web* seperti HTML, CSS, JavaScript, dan AJAX. Tetapi menawarkan pengalaman yang mirip dengan sebuah aplikasi kepada penggunaannya. Seringnya aplikasi *web mobile* juga memanfaatkan fitur-fitur HTML5, seperti akses *offline* dan geolokasi untuk menyediakan pengalaman yang lebih baik (Wahana Komputer, 2013).

2.3 Alat Bantu Dalam Analisa dan Perancangan Sistem

2.3.1 Data Flow Diagram (DFD)

DFD merupakan diagram yang digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana *data* tersebut mengalir (misalnya lewat telpon, surat dan sebagainya) atau lingkungan fisik dimana *data* tersebut akan disimpan.

DFD merupakan alat yang digunakan pada metodologi pengembangan sistem yang terstruktur (*structured Analysis and design*). DFD merupakan alat yang

cukup populer sekarang ini, karena dapat menggambarkan arus *data* di dalam sistem dengan terstruktur dan jelas. Lebih lanjut DFD juga merupakan dokumentasi dari sistem yang baik. Beberapa simbol yang digunakan di DFD untuk maksud mewakili:

1. *External Entity*

Setiap sistem pasti mempunyai batas sistem (*boundary*) yang memisahkan suatu sistem dengan lingkungan luarnya. Sistem akan menerima *input* dan menghasilkan *output* kepada lingkungan luarnya. Kesatuan luar (*external entity*) merupakan kesatuan (*entity*) di lingkungan luar sistem yang dapat berupa orang, organisasi atau sistem lainnya yang berada di lingkungan luarnya yang akan memberikan *input* atau menerima *output* dari sistem.

2. *Data Flow*

Arus data (*data flow*) di DFD diberi simbol suatu panah. Arus data ini mengalir diantara proses (*process*), simpanan data (*data store*) dan kesatuan luar (*external entity*). Arus data ini menunjukkan arus dari data yang dapat berupa masukan untuk sistem atau hasil dari proses sistem.

3. *Process*

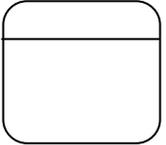
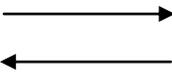
Suatu proses adalah kegiatan atau kerja yang dilakukan oleh orang, mesin atau komputer dari hasil suatu arus data yang masuk ke dalam proses untuk dihasilkan arus data yang akan keluar dari proses. Untuk *physical data flow diagram (PDFD)*, proses dapat dilakukan oleh orang, mesin, atau komputer, sedangkan untuk *logical data flow diagram (LDFD)*,

suatu proses hanya menunjukkan proses dari komputer. Suatu proses dapat ditunjukkan dengan simbol lingkaran atau dengan simbol empat persegi panjang tegak dengan sudutnya yang tumpul.

4. *Data Store*

Simpanan data di DFD dapat disimbolkan dengan sepasang garis horisontal paralel yang tertutup di salah satu ujungnya. Nama dari *data store* menunjukkan nama dari filenya, misal file langganan, file hutang, file arsip faktur dan lain sebagainya. Untuk PDFD, supaya memperjelas simpanan data ini, penjelasan mengenai media dari simpanan data perlu dicantumkan seperti misalnya buku atau arsip atau suatu kotak dan lain sebagainya. Sedangkan untuk LDFD, penjelasan ini dapat digunakan untuk identifikasi dari simpaan data yang berguna sebagai acuan dalam merancang database (Jogiyanto, 2005).

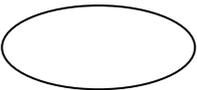
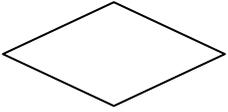
Tabel 2.1 Simbol *Data Flow Diagram*

Simbol	Nama	Fungsi
	Simbol Entitas Eksternal	Digunakan untuk menunjukkan tempat asal <i>data</i> atau sumber atau tempat tujuan <i>data</i> atau tujuan.
	Simbol Proses	Digunakan untuk menunjukkan tugas atau proses yang dilakukan baik secara manual atau otomatis
	Simbol Penyimpanan Data	Digunakan untuk menunjukkan arus dari proses.
	Simbol Arus Data	Digunakan untuk menunjukkan arus dari proses.

2.3.2 Entity Relationship Diagram (ERD)

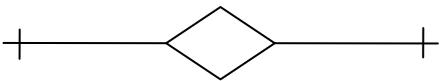
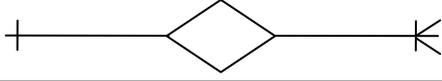
ERD merupakan diagram yang menggambarkan keterkaitan antar entitas yang satu dengan entitas yang lain dalam suatu sistem yang terintegrasi. Adapun simbol-simbol dalam diagram relasi ERD dapat dilihat pada tabel 2.2 berikut ini.

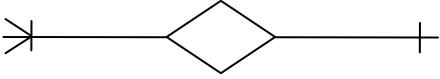
Tabel 2.2 Simbol *Entity Relationship Diagram*

Simbol	Nama	Keterangan
	Simbol <i>Entity</i>	Suatu <i>entity</i> merupakan suatu objek atau konsep mengenai tempat yang anda inginkan untuk menyimpan informasi
	Simbol Atribut	Atribut adalah sifat-sifat atau karakteristik suatu entitas
	Simbol Relasi	Relasi mengilustrasikan bagaimana dua entitas terbagi informasi didalam struktur basis data

Konektivitas dari suatu hubungan menguraikan pemetaan dari kejadian entitas yang dihubungkan. Jenis dasar konektivitas untuk hubungan adalah satu-ke-satu, satu-ke-banyak, dan banyak-ke-banyak. Adapun simbol kardinalitas dalam *entity relationship diagram* dapat dilihat pada tabel 2.3 berikut ini (Yakub, 2012).

Tabel 2.3 Simbol Kardinalitas Dalam *Entity Relationship Diagram*

Simbol yang digunakan	Jenis Kerelasian
	1-ke-1
	1-ke-n

	n-ke-1
	n-ke-n

2.3.3 Flowchart

Flowchart adalah bagan (*chart*) yang menunjukkan alir (*flow*) di dalam program atau prosedur sistem secara logika. Bagan alir digunakan terutama untuk alat komunikasi dan untuk dokumentasi. Pada waktu akan menggambar suatu bagan alir, analis sistem atau pemrogram dapat mengikuti pedoman-pedomain sebagai berikut

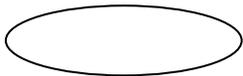
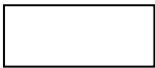
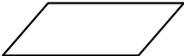
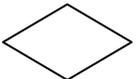
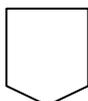
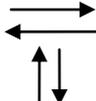
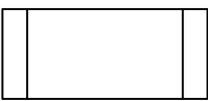
1. Bagan alir sebaiknya digambar dari atas ke bawah dan mulai dari bagian kiri dari suatu halaman
2. Kegiatan di dalam bagan alir harus ditunjukkan dengan jelas
3. Harus ditunjukkan dari mana kegiatan akan dimulai dan dimana akan berakhir
4. Masing-masing kegiatan di dalam bagan alir sebaiknya digunakan suatu kata yang mewakili suatu pekerjaan
5. Masing-masing kegiatan di dalam bagan alir harus di dalam urutan yang semestinya
6. Kegiatan yang terpotong dan akan disambung ditempat lain harus tunjukkan dengan jelas menggunakan simbol penghubung
7. Gunakan simbol-simbol bagan alir yang standar

Selain itu *flowchart* digunakan untuk mempermudah penyusunan program.

Dengan menggunakan *flowchart*, logika pemrograman lebih mudah dipahami dan

dianalisis, sehingga anda dapat menentukan kode-kode pemrograman yang sesuai dengan pekerjaannya. *Flowchart* program dapat disamakan dengan *blue print* bangunan. Seperti diketahui arsitek akan membuat *blue print* bangunan sebelum memulai konstruksinya. Demikian pula seorang *programmer* disarankan untuk membuat *flowchart*. Sebelum menulis kode programnya. Berikut beberapa simbol standar *flowchart* yang sering digunakan dalam pemrograman komputer (Jogiyanto, 2005).

Tabel 2.4 Simbol Program *Flowchart*

Simbol	Arti
	Simbol <i>Start</i> atau <i>End</i> yang mendefenisikan awal atau akhir dari sebuah <i>flowchart</i>
	Simbol pemrosesan yang terjadi pada sebuah alur kerja
	Simbol <i>Input/output</i> mendefenisikan masukan dan keluaran proses
	Simbol untuk memutuskan proses lanjutan dari kondisi tertentu
	Simbol konektor untuk menyambung proses pada lembar kerja yang berbeda
	Simbol konektor untuk menyambung proses lembar kerja yang berbeda
	Simbol untuk menghubungkan antar proses atau antar simbol
	Simbol yang menyatakan bagian dari program (sub program)

2.4 Alat Bantu Dalam Pembuatan Sistem

2.4.1 *JQuery Mobile*

JQuery Mobile adalah *web framework* yang dioptimalkan untuk perangkat-perangkat layar sentuh (*touch-optimized*). Sebagai library, *JQuery Mobile* yang juga dikenal sebagai *library JavaScript* maupun *framework mobile* ini dikembangkan oleh tim jQuery Project. Tujuan pengembangannya adalah berfokus pada pembuatan *framework* yang *kompetible* dengan beraneka ragam perangkat *smartphone* dan *tablet*. *JQuery Mobile* dikembangkan sebagai alternatif yang menjawab kebutuhan akan pengembangan aplikasi *web mobile* yang mudah dan memiliki banyak potensi. *JQuery Mobile* sangat memungkinkan untuk membuat aplikasi *web mobile* yang penampilan dan interaktivitasnya konsisten di semua perangkat yang mendukung, serta memiliki kemampuan *user interface* (UI) yang canggih.

JQuery mobile merupakan sistem *user interface* berbasis HTML5 dan pengembangan dari *library jQuery UI* yang sedang populer saat ini. Kelebihan jQuery adalah kodennya ringan, menerapkan desain yang dapat beradaptasi dengan berbagai jenis perangkat mobile dan menonjolkan markup semantik. Oleh karena itu, desainnya sangat fleksibel dan mempunyai *thema* yang bisa disesuaikan. Hal ini membuat *JQuery Mobile* sangat sekali digunakan. Adapun kemampuan atau fitur yang dimiliki *JQuery Mobile* antara lain sebagai berikut

1. *Framework* yang ideal
2. Menghasilkan *user interface* yang konsisten
3. Bersifat *markup-driven* yang mempermudah pengembangan
4. Bebas membuat *theme* (Wahana Komputer, 2013)

2.4.2 HTML

HTML merupakan kependekan dari *Hypertext Markup Language*. Sebuah file HTML merupakan sebuah file teks yang berisi tag-tag markup. Tag markup memberitahukan browser bagaimana harus menampilkan sebuah halaman. File HTML harus memiliki ekstensi htm atau html, dan file HTML dapat dibuat menggunakan editor teks (notepad) yang biasa Anda pakai (Prasetio, 2014).

2.4.3 PHP

PHP (*hypertext preprocessor*) merupakan bahasa script yang ditanam di sisi server. PHP dapat melakukan apa saja yang bisa dilakukan program CGI lain, yaitu mengolah *data* dengan tipe apapun, menciptakan halaman web yang dinamis, serta menerima dan menciptakan cookies. PHP dapat berjalan pada semua jenis sistem operasi antara lain pada Linux dan varian Unix (HP-UX, Solaris dan OpenBSD), pada Ms Windows, Mac dan masih banyak lagi, selain itu PHP juga dapat berjalan pada beberapa jenis web-server antara lain Apache, Microsoft IIS, personal webserver, Netscape dan lain sebagainya. Adapun keunggulan dari PHP adalah sebagai berikut:

1. Kesederhanaan. Buat Anda yang baru belajar pemrograman alasan ini merupakan alasan pertama untuk mulai belajar PHP. Umumnya bahkan yang sedikit tahu atau bahkan sama sekali yang tidak tahu tentang pemrograman bisa dengan cepat dan mulai membuat aplikasi lengkap. Hal ini memungkinkan karena PHP khusus didesain untuk membuat aplikasi web dan karena itu, PHP memiliki banyak sekali fungsi build in untuk memenuhi kebutuhan standar aplikasi web.

2. PHP bersifat open source, karena *source code* PHP tersedia secara gratis, maka hal itu memungkinkan komunitas *developer* untuk melakukan perbaikan, pengembangan, dan menemukan *bug* dalam bahasa PHP.
3. Stabilitas dan kompatibilitas. Saat ini PHP berjalan dengan stabil diberbagai macam sistem operasi seperti berbagai versi UNIX (Linux), Windows, dan Mac. PHP juga terintegrasi secara baik dengan berbagai macam web server termasuk dua yang terpopuler ISS dan Apache.
4. PHP juga dilengkapi dengan pendukung lain seperti support langsung ke berbagai macam database yang populer, arsitektur yang dapat dikembangkan dan prosesor yang tidak hanya menggunakan resource minimal komputer Anda dibanding kompetitornya, tetapi juga dapat menampilkan halaman *web* dengan cepat (Prasetio, 2014).

2.4.4 JavaScript

JavaScript merupakan bahasa pemrograman berbasis client, artinya bahasa ini berjalan pada sisi browser pengguna, bukan pada *server*. Salah satu keuntungan dari bahasa ini adalah ringan karena berjalan pada masing-masing *browser* dan pekerjaan tidak dibebankan pada server. Jika diakses oleh 1000 orang maka setiap beban akan ditanggung oleh masing-masing browser, bukan server yang menanggung eksekusi 1000 pengakses tersebut.

Pada awalnya, *JavaScript* digunakan dalam dunia *web* untuk menangani berbagai *event* pengguna dan melakukan validasi. Menangani *event* pengguna misalnya menangani ketika pengguna mengubah pilihan pada *combo box* sehingga sekarang dikenal dengan istilah AJAX, yang memanfaatkan *JavaScript*

sebagai teknologi utama. Begitu juga dengan validasi pada halaman *website*. Jika validasi dapat dilakukan oleh *JavaScript* tentu akan efisien karena tidak perlu dilakukan dan dikirimkan ke server. Contoh yang sering dipakai adalah untuk mendeteksi inputan kosong, cukup dilakukan oleh *JavaScript* pada *browser* masing-masing pengguna (Nugroho, 2012).

2.4.5 CSS

CSS (*Cascading Style Sheet*) merupakan bahasa *scripting* yang sangat ringan dan mudah dipelajari. CSS biasanya digunakan untuk menentukan tema tampilan dari sebuah *website*. Dengan menggunakan CSS, kita dapat menentukan tampilan suatu elemen yang sama pada setiap halaman sehingga kita tidak perlu melakukan penulisan kode program secara berulang-ulang untuk menampilkan tampilan yang sama (Nugroho, 2012).

2.4.6 MySQL

MySQL adalah sebuah *server database open source* yang paling populer saat ini. MySQL umumnya digunakan bersamaan dengan *script PHP* untuk membuat aplikasi *server* yang dinamis dan *powerful*. MySQL mampu menerima dan mengirimkan datanya dengan sangat cepat, multi user serta menggunakan standar SQL (*Structured Query Language*)". Dengan menggunakan MySQL server maka data dapat diakses oleh banyak pemakai secara bersamaan sekaligus dapat membatasi akses para pemakai berdasarkan *privillage* (hak user) yang diberikan. MySQL menggunakan bahasa SQL (*Structure Query Language*)

yaitu bahasa standar pemrograman database. Adapun keunggulan dari MySQL adalah sebagai berikut:

1. Bersifat *open source*, artinya program ini bersifat *free* atau bebas digunakan oleh siapa saja tanpa harus membeli dan membayar lisensi kepada pembuatnya.
2. MySQL merupakan sebuah *database server*, jadi dengan menggunakan *database* ini dapat menghubungkannya ke media internet sehingga dapat diakses dari jauh.
3. MySQL merupakan sebuah *database* yang mampu menyimpan data berkapasitas sangat besar hingga berukuran *Gigabyte* sekalipun.
4. Sistem *software*-nya tidak memberatkan kerja dari *server* atau komputer, karena dapat bekerja di *background*.
5. MySQL didukung oleh driver ODBC, artinya database MySQL dapat diakses menggunakan aplikasi apa saja termasuk berupa visual seperti Delphi maupun Visual Basic.
6. Cukup aman karena memiliki *password* untuk mengaksesnya.
7. Mendukung *field* yang dijadikan sebagai kunci primer dan kunci unik (Prasetio, 2014).