



BAB II

TINJAUAN PUSTAKA

2.1. Teori Umum

2.1.1. Pengertian Sistem Informasi

Menurut Jogiyanto (2010:697), menjelaskan bahwa sistem informasi dapat didefinisikan sebagai suatu sistem di dalam suatu organisasi yang merupakan kombinasi dari orang-orang, fasilitas, teknologi, media, prosedur-prosedur dan pengendalian yang ditunjukkan untuk mendapatkan jalur komunikasi penting, memproses tipe transaksi rutin tertentu, memberi sinyal kepada manajemen dan yang lainnya terhadap kejadian-kejadian internal dan eksternal yang penting dan menyediakan suatu dasar informasi untuk pengambilan keputusan yang cerdas.

Menurut Sutabri (2012:38), juga berpendapat bahwa sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan strategi dari suatu organisasi untuk dapat menyediakan laporan-laporan yang diperlukan oleh pihak luar tertentu.

Dari definisi diatas maka penulis menyimpulkan bahwa sistem informasi merupakan sekumpulan prosedur di dalam organisasi yang pada saat dilaksanakan akan memberikan informasi bagi pengambil keputusan.

2.1.2. Pengertian *Monitoring*

Monitoring adalah proses pengumpulan dan analisis informasi (berdasarkan indikator yg ditetapkan) secara sistematis dan kontinu tentang kegiatan program/proyek sehingga dapat dilakukan tindakan koreksi untuk penyempurnaan program/proyek itu selanjutnya (Hikmat, 2012:2).

Menurut Mardiani (2013:36), *monitoring* adalah proses rutin pengumpulan data dan pengukuran kemajuan atas objektif program. Memantau perubahan yang fokus pada proses dan keluaran. *Monitoring* menyediakan data mentah untuk menjawab pertanyaan sedangkan evaluasi adalah meletakkan data-data tersebut agar dapat digunakan dan dengan demikian memberikan nilai tambah.



Sehingga penulis menyimpulkan bahwa *monitoring* adalah penilaian secara terus menerus terhadap fungsi kegiatan-kegiatan program-program di dalam hal jadwal penggunaan input/masukan data oleh kelompok sasaran berkaitan dengan harapan-harapan yang telah direncanakan.

Adapun menurut Mardiani (2013:37), tujuan dilakukannya *monitoring* diantaranya adalah sebagai berikut:

1. Mengkaji apakah kegiatan-kegiatan yang dilaksanakan telah sesuai dengan rencana.
2. Mengidentifikasi masalah yang timbul agar langsung dapat diatasi.
3. Melakukan penilaian apakah pola kerja dan manajemen yang digunakan sudah tepat untuk mencapai tujuan kegiatan.
4. Mengetahui kaitan antara kegiatan dengan tujuan untuk memperoleh ukuran kemajuan.
5. Menyesuaikan kegiatan dengan lingkungan yang berubah, tanpa menyimpang dari tujuan.

2.1.3. Pengertian Evaluasi

Evaluasi adalah proses penilaian pencapaian tujuan dan pengungkapan masalah kinerja program/proyek untuk memberikan umpan balik bagi peningkatan kualitas kinerja program/proyek. (Hikmat, 2012:2)

Menurut Mardiani (2013:36), evaluasi adalah tempat belajar kejadian, pertanyaan yang perlu dijawab, rekomendasi yang harus dibuat, menyarankan perbaikan. Namun tanpa *monitoring*, evaluasi tidak akan ada dasar. Tidak memiliki bahan baku untuk bekerja dan terbatas pada wilayah spekulasi oleh karena itu *monitoring* dan evaluasi harus berjalan seiring.

Dari beberapa pengertian diatas dapat disimpulkan bahwa evaluasi merupakan proses penilaian terhadap hasil *monitoring* suatu program/proyek dalam suatu periode untuk digunakan sebagai bahan peningkatan kualitas program/proyek selanjutnya.



2.1.4. Pengertian *Dashboard*

Dashboard adalah alat penting untuk memonitor keadaan sehari-hari suatu organisasi dan sebagai alat penghubung tunggal yang digunakan untuk membuat keputusan dan mempunyai akses untuk mengunci indikator kinerja (KPIs). *Dashboard* seperti sebuah jaringan lokal yang eksekutif, dimana semua kegiatan ditampilkan (Soleh dkk, 2013:15).

Dashboard adalah sebuah alat yang dapat menampilkan informasi penting secara visual untuk mendukung tujuan organisasi, dapat menggabungkan informasi pada sebuah layar sehingga dapat dilihat secara sekilas. Tampilan visual disini didefinisikan bahwa penyajian informasi harus dirancang sebaik mungkin, agar memudahkan pengguna untuk menangkap serta memahami makna informasi secara cepat dan tepat (Few, 2006:26).

(Imelda, 2013:119) *Dashboard* akan memonitoring berbagai informasi yang relevan dan dibutuhkan oleh suatu organisasi atau perusahaan secara real time dengan bermacam-macam format seperti *graphical gadgets*, *typically gauges*, *charts*, *indicators*, dan *colorcoded maps* yang mendukung pengguna dalam pengambilan keputusan secara cepat dan cerdas.

Jadi, *dashboard* merupakan salah satu solusi dalam penyajian dan visualisasi data serta suatu model antarmuka sistem informasi yang dianalogikan seperti dashboard sebuah mobil yang mudah dipelajari. *Dashboard* dapat mengkomunikasikan informasi penting dengan cepat. Sebuah desain yang baik untuk penyajian dan visualisasi data akan memberikan kejelasan mengenai informasi penting yang disampaikan bagi pengguna. Dengan desain yang baik, *dashboard* yang dibuat akan dapat membantu dalam mengidentifikasi tren, pola dan anomali pada data sehingga pada akhirnya dapat membantu dalam pengambilan keputusan yang efektif dan efisien.

Teori khusus menjelaskan mengenai pengertian yang berkaitan dengan Pemrograman Berorientasi Objek, Diagram *Unified Modeling Language* (UML) berupa *Usecase Diagram*, *Activity Diagram*, *Squence Diagram* dan *Class Diagram* beserta simbol-simbol yang digunakan. Sedangkan teori program berkaitan dengan program yang digunakan seperti Bahasa Pemrograman Berbasis



Web (*PHP*), *database MySQL* dan teori lain yang dijadikan sebagai acuan pembahasan.

Terdapat tiga jenis *dashboard* berdasarkan tingkat manajemen yang didukung, yaitu:

1. *Strategic dashboard*

Mendukung manajemen level strategis untuk memberikan informasi yang dapat digunakan dalam pembuatan keputusan bisnis, memprediksi peluang dan memberikan informasi pencapaian tujuan strategis.

2. *Tactical dashboard*

Mendukung manajemen level taktikal. Digunakan untuk memberikan informasi yang diperlukan untuk mengetahui penyebab suatu kejadian. Fokus dengan proses analisis untuk menentukan penyebab dari suatu masalah. Memiliki konten informasi yang lebih banyak, didesain untuk berinteraksi dengan data seperti *drill-down* dan tidak memerlukan data.

3. *Operational dashboard*

Mendukung manajemen level operasional. Memberikan informasi tentang kegiatan yang sedang terjadi serta perubahan *real-time*. Sangat bermanfaat dalam memberikan kewaspadaan pada hal-hal yang perlu direspon secara cepat. Fokus terhadap monitoring aktivitas dan kejadian yang berubah secara konstan. Informasi yang disajikan dengan sangat spesifik dan sangat detail, dinamis dan memerlukan data *real-time*.

2.1.5. Pengertian *Key Performance Indicator (KPI)*

Menurut Parmenter (2007), *Key Performance Indicators (KPI)* merupakan sekumpulan ukuran yang berfokus pada aspek-aspek dari kinerja organisasi yang paling kritis untuk saat ini dan keberhasilan di masa depan organisasi. KPI harus dipantau setiap saat, setiap hari, atau mungkin mingguan. Pelaporan pengukuran untuk manajemen harus tepat waktu. KPI perlu dilaporkan “24/7”, setiap hari, atau mingguan. Sedangkan pengukuran kinerja lainnya (bukan utama) dapat dilaporkan secara berkala, bulanan atau triwulanan.



Key Performance Indicators atau KPI adalah salah satu metode yang dapat digunakan untuk mengukur kinerja perusahaan secara keseluruhan atau kinerja tiap-tiap bagian dalam perusahaan (Kurniawan 2017:10).

(Kurniawan 2017:11) KPI dapat disebut juga sebagai Indikator Kinerja Utama (IKU). Masing-masing organisasi tentu mempunyai IKU yang berbeda-beda dan unik berdasarkan jenis, sifat, tujuan dan strateginya. Contohnya institusi pendidikan tinggi menetapkan jumlah yang lulus, Index prestasi akademik, lulusan yang dapat diserap pasar kerja, jumlah hasil penelitian yang memiliki hak paten, jurnal ilmiah setiap tahunnya, tingkat akreditasi nasional dan seterusnya sebagai IKU. Contoh lain dalam dunia bisnis, IKU antara lain perkembangan kinerja karyawan, jumlah dan kualitas produksi, omset penjualan, dan keuntungan persatuan waktu.

Oleh karena itu, KPI memiliki peran penting dalam kemajuan suatu organisasi. Dengan adanya KPI organisasi dapat mengukur dan mengetahui pencapaian kerjanya. Sehingga akan menuntut organisasi mempunyai visi dan misi yang jelas, dan sebagai penentu langkah praktis untuk mencapai tujuannya.

KPI di suatu perusahaan memungkinkan terjadi perubahan, seiring dengan perkembangan atau kebutuhan bisnis dan strategi yang dijalankan perusahaan. Sehingga pengembangan KPI harus diupayakan secara kontinyu untuk menjamin kesesuaian kinerja dengan tujuan dan strategi organisasi.

Untuk menghitung skor KPI yaitu dengan rumus sederhana Pencapaian Aktual KPI (A) dibagi Target KPI (T) dikalikan seratus, atau menggunakan rumus kebalikannya Target KPI (T) dibagi Pencapaian Aktual KPI (A) dikalikan seratus.

$$\text{Skor KPI} = (A/T) \times 100 \dots\dots\dots (1)$$

$$\text{Skor KPI} = (T/A) \times 100 \dots\dots\dots (2)$$



2.2. Teori Khusus

2.2.1. Pemograman Berorientasi Objek

2.2.1.1. Pengertian Pemograman Berorientasi Objek

Nugroho (2005:6), menjelaskan bahwa paradigma berorientasi objek adalah cara yang berbeda dalam memandang aplikasi-aplikasi. Dengan pendekatan berorientasi objek, para pengembang membagi aplikasi-aplikasi besar menjadi objek-objek, yang mandiri satu terhadap yang lainnya.

Sukanto dan Shalahuddin (2013:100), menjelaskan bahwa berorientasi objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya.

Dari beberapa definisi diatas dapat disimpulkan bahwa pemograman berbasis objek adalah kumpulan objek yang dapat mengorganisasikan perangkat lunak.

2.2.1.2. Ciri Pemograman Berorientasi Objek

Siallagan (2009:149), menjelaskan bahwa ciri-ciri atau karakteristik pemograman berorientasi objek, antara lain:

a. Abstraksi (*Abstraction*)

Abstraksi adalah pengabstrakan atau melakukan seleksi terhadap aspek-aspek tertentu suatu masalah. Abstraksi digunakan untuk menyembunyikan kerumitan dari suatu proses. Sebagai contoh, dalam membuat suatu sistem, ada tombol-tombol yang dapat digunakan. Operator atau pengguna tidak perlu berpikir tentang pembuatan tombol tersebut, tetapi yang penting mereka dapat menggunakannya.

b. Pembungkusan (*Encapsulation*)

Pembungkusan sering pula disebut pengkapsulan. Artinya, proses membuat paket (memaketkan) data objek bersama dengan metode-metodenya. Berdasarkan kode program, proses memisahkan aspek-aspek objek dilakukan dengan pembungkusan. Proses pembungkusan itu sendiri merupakan cara atau mekanisme untuk melakukan abstraksi.



c. Pewarisan (*Inheritance*)

Pewarisan adalah memberikan atau mewariskan sesuatu kepada keturunan berikutnya. Misalnya, seorang anak pasti akan mewarisi beberapa sifat atau perilaku yang dimiliki oleh ibu/bapaknya. Dalam konteks ini, suatu kelas dalam program dapat diturunkan menjadi kelas-kelas baru lainnya yang akan mewarisi beberapa sifat atau perilaku dari kelas induknya.

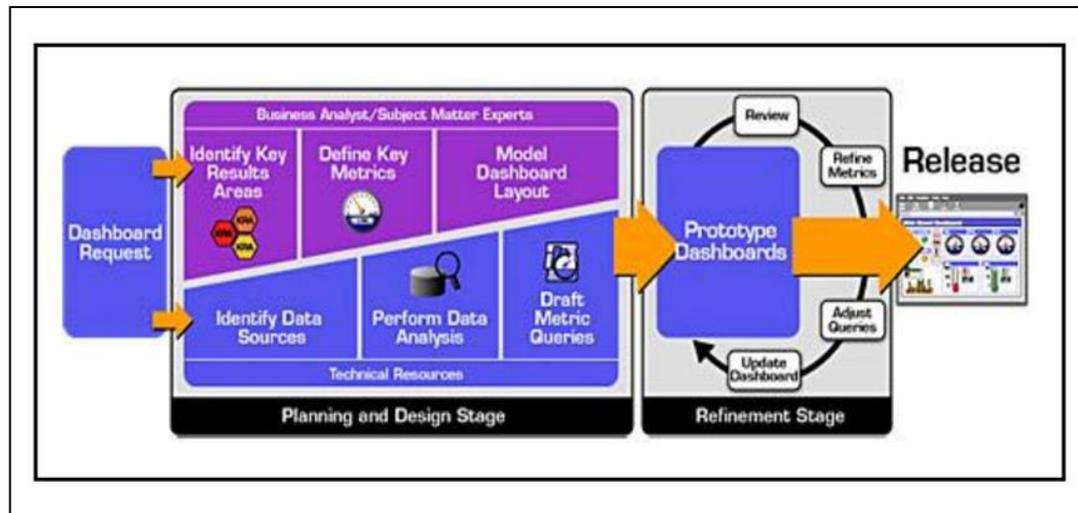
d. Polimorfisme (*Polymorphism*)

Polimorfisme adalah suatu kejadian ketika objek dapat mengungkap banyak hal melalui satu cara yang sama.

2.2.2. Metodologi *Pureshare*

Metode yang akan penulis digunakan dalam pengembangan *dashboard* sistem ini adalah metode *Pureshare*. Metode *Pureshare* adalah salah satu metode pembangunan *dashboard* yang dikembangkan oleh vendor *Pureshare* untuk memberi fasilitas terhadap proyek yang berkaitan dengan usaha pengelolaan dan pengukuran kinerja organisasi, termasuk pembangunan *dashboard*. Pembangunan *dashboard* dirancang supaya selaras dengan kebutuhan teknologi dan tujuan bisnisnya. Metode *pureshare* menggunakan dua pendekatan yang biasa disebut *top-down design* dan *bottom-up implementation*, yang digambarkan pada gambar 2.1. Metode ini melibatkan pengguna secara aktif dalam proyek pembangunan *dashboard* secara cepat. Proses ini terbukti menurunkan tingkat resiko proyek dengan melibatkan *end user* dalam pembuatan *dashboard* serta mempercepat dalam penerapannya. (Haryanti dalam Dewi, Widiyanto, 2015:164).

Gambaran umum dari kerangka metode yang dikembangkan oleh *Pureshare* dapat dilihat pada gambar dibawah ini.



Sumber: (Dewi dan Widiyanto, 2015:164)

Gambar 2.1 Tahapan Metode *Pureshare*

(Dewi dan Widiyanto, 2015:164) Adapun tahapan pengembangan sistem menggunakan metode *Pureshare*, yaitu:

1. Perencanaan dan Desain

Tahapan ini dipimpin oleh analis bisnis. Perancangan dan desain dilakukan dengan pendekatan *top-down* dan fokus terhadap kebutuhan bisnis pengguna. Tujuan rancang bangun dashboard mengacu pada *Key Result Area* (KRA) yang digunakan untuk mengidentifikasi *Key Performance Indicators* (KPI). KPI dikategorikan sebagai *alert*, *monitor*, atau *result/trends*. Pada tahap akhir, perancang mengidentifikasi kebutuhan desain dashboard kemudian merancang model hirarki dashboard dan desain layout dashboard. Perancang harus memahami kebutuhan pengguna serta mampu berkomunikasi dengan pengguna untuk menentukan fitur-fitur kunci yang ditampilkan pada dashboard.

2. Review Sistem dan Data

Tahap ini dilakukan bersamaan dengan tahap perencanaan dan desain. Pendekatan *bottom-up implementation* dilakukan pada tahap ini. Menentukan data apa yang tersedia, bagaimana mengaksesnya, menilai kualitas, dan



mengembangkan metrik. Menganalisa data untuk lebih memahami sumber data dan mengembangkan rekomendasi.

3. Perancangan Prototipe

Pembuatan prototipe antarmuka untuk memberikan pandangan singkat dari hasil akhir yang didapatkan. Dilakukan dengan pendekatan *top-down* dan *bottom-up* secara bersamaan. Pada tahap ini pengguna terlibat untuk memberikan umpan balik dan saran untuk implementasi.

4. Perbaikan Prototipe

Melakukan serangkaian ulasan prototype. Mengumpulkan umpan balik dan membuat penyesuaian yang diperlukan. Memastikan bahwa informasi yang digunakan tepat dan data yang ada akurat. Menentukan ambang batas yang tepat dan target.

5. Rilis

Melatih pengguna sehingga dapat dengan mudah menggunakan metrik dan menilai informasi yang disampaikan.

6. Perbaikan Terus Menerus

Metodologi ini memfasilitasi siklus perbaikan secara terus menerus. Tahapan ini dapat digunakan ulang untuk rancang bangun dashboard di berbagai area organisasi.

2.2.3. Beasiswa Insan Madani Dompot Dhuafa Sumatera Selatan

2.2.3.1. Sekilas Tentang Beasiswa Insan Madani

Program Beasiswa Insan Madani sebelumnya bernama Beasiswa Insan Mulia merupakan salah satu program Dompot Dhuafa Sumatera Selatan di bidang pendidikan. Dibuka sejak bulan Februari 2017, lebih kurang 100 orang siswa SMA/SMK/MA di kota Palembang telah merasakan manfaat program ini. Di awal program sebanyak sepuluh tutor ikut berkontribusi mensukseskan program ini. Tidak hanya memberikan beasiswa berupa biaya pendidikan, program ini disusun dengan kurikulum khusus untuk membekali siswa agar memiliki wawasan tentang ilmu agama dan ilmu pengetahuan yang mumpuni.



Para siswa ini kemudian disebut sebagai penerima manfaat. Para penerima manfaat program Beasiswa Insan Madani merupakan anak-anak prestasi di sekolah yang masuk dalam kategori dhuafa. Mereka harus mengikuti serangkaian proses seleksi program, mulai dari seleksi berkas administrasi, tes psikologi dan wawancara hingga tahap *home visit* untuk memverifikasi kondisi rumah dan keluarganya. Setelah dinyatakan lulus, para penerima manfaat ini wajib memenuhi komitmen untuk mengikuti semua kurikulum yang telah disusun selama satu tahun. Setiap akhir tahun program akan ada evaluasi terhadap komitmen masing-masing penerima manfaat dalam mengikuti program apakah diteruskan di tahun berikutnya atau tidak.

2.2.3.2. Pengertian Penerima Manfaat

Dalam arti luas penerima manfaat atau *beneficiary* adalah seseorang atau badan hukum yang menerima manfaat dari *benefactor* (sang penolong) (Wikipedia, 2010).

Dalam *Collins English Dictionary-Complete and Unabridged* (HaperCollins Publisher, 2003), makna dari penerima manfaat adalah:

- a. Arti luas: “*a person who gains or benefit in some way from something*”: seseorang yang mendapat keuntungan atau manfaat dalam beberapa hal dari segala sesuatu.
- b. Dalam kesejahteraan sosial: “*a person who receives government assistance social security beneficiary*” : seseorang yang mendapatkan uang jaminan sosial dari bantuan pemerintah.

2.2.3.3. Mengukur *Performance* Penerima Manfaat

Dalam menjalankan kurikulum program Beasiswa Insan Madani terdapat indikator atau kriteria yang digunakan untuk mengukur *performance* penerima manfaat berdasarkan standar lembaga. Berikut adalah kurikulum program tahun 2018 yang digunakan untuk mengukur *performance* penerima manfaat:

1. Absensi Kegiatan Luar Jaringan (LARING)

Adalah nilai kehadiran pada agenda pengembangan diri yang diadakan



melalui pertemuan tatap muka setiap bulan. Skala penilaian untuk kriteria ini adalah 0 sampai dengan 4, dimana nilai 4 = hadir, 2 = izin/sakit dan 0 = tanpa keterangan.

2. Keaktifan dalam Kegiatan

Merupakan penilaian selama kegiatan laring berlangsung. Tutor wajib memberikan perhatian ketika kegiatan tatap muka berlangsung, biasanya penilaian ini berdasarkan keaktifan penerima manfaat dalam menanggapi materi, menjawab maupun bertanya ketika pemateri selesai menyampaikan materinya. Skala penilaiannya adalah 0 dan 1, dimana 0 = tidak aktif sedangkan 1 berarti aktif.

3. *Reward*

Setiap pertemuan disediakan *reward* untuk memberi semangat para penerima manfaat, macam-macam *reward* tergantung tema kegiatan. Sama seperti penilaian keaktifan skala penilaian *reward* pun 0 dan 1, dimana 1 = mendapat *reward* dan 0 = tidak mendapat *reward*.

4. Resensi Buku

Membaca minimal dua buah buku merupakan hal yang wajib dilaksanakan para penerima manfaat setiap bulannya. Hal ini merupakan salah satu cara membangkitkan semangat literasi para penerima manfaat. Untuk itu disediakan sebuah lembar untuk menuliskan *resensi* atau kesimpulan dari buku yang sudah mereka baca. Adapun skala penilaiannya adalah 0 sampai dengan 4, dimana 4 = meresensi dua buah buku, 2 = meresensi sebuah buku, dan 0 = tidak meresensi buku.

5. Resensi Berita

Selain membaca buku, para penerima manfaat harus *update* mengenai informasi-informasi terkini melalui berita baik dari media cetak maupun elektronik. Setiap bulan ada empat berita yang harus mereka resensi, oleh karena itu skala penilaiannya pun 0 hingga 4. Penilaian berdasarkan berapa jumlah resensi berita yang mereka tulis, nilai 4 untuk resensi empat berita. Nilai 3 = resensi tiga berita, 2 = resensi dua berita 1 = resensi satu berita dan 0 = tidak menulis resensi berita.



6. Pembinaan Penerima Manfaat Tidak Langsung

Sebuah kegiatan yang dirancang untuk melatih para penerima manfaat agar mau berbagi ilmu pada orang lain, baik teman sebaya, adik kelas maupun yang lebih tua. Standar kegiatan ini adalah satu penerima manfaat minimal memiliki satu binaan dan melakukan dua kali pertemuan setiap bulan. Skala penilaiannya adalah 0 hingga 2, 2 = membina satu atau lebih orang dengan dua atau lebih pertemuan tiap bulan, 1 = membina satu atau lebih orang dengan satu kali pertemuan tiap bulan, dan 0 = tidak melakukan pembinaan.

7. Absensi Diskusi Dalam Jaringan (DARING)

Adalah penilaian atas kehadiran kegiatan dalam jaringan (daring) atau biasa disebut Mentoring Online (Mentol). Sama seperti absensi kegiatan di luar jaringan (laring), skala penilaian kriteria ini adalah 0 hingga 4.

8. Pengiriman Kesimpulan Diskusi

Setelah selesai mentol, para penerima manfaat juga wajib mengirimkan kesimpulan masing-masing mengenai materi yang sudah dibahas ketika mentol. Adapun penilaiannya adalah 0 dan 1, dimana 1 berarti mengirimkan kesimpulan sedang 0 berarti tidak mengirimkan kesimpulan.

Dalam menghitung skor tiap-tiap kriteria digunakan rumus sebagai berikut:

$$\text{Skor} = \frac{\text{Nilai yang didapat}}{\text{Nilai maksimal}} \times \text{bobot}$$

Setelah mendapatkan skor masing-masing kriteria, selanjutnya adalah menjumlahkan seluruh skor untuk mendapatkan skor akhir untuk melihat *performance* (kinerja) masing-masing penerima manfaat. Skor akhir minimal yang harus didapat adalah 70 dari skala 100, jika penerima manfaat mendapatkan skor akhir dibawah standar maka akan dilakukan evaluasi terhadap masing-masing penerima manfaat. Periode evaluasi dilakukan setiap bulan oleh penanggung jawab program.



2.2.4. Unified Modelling Language (UML)

2.2.4.1. Pengertian Unified Modelling Language (UML)

Prabowo dan Herlawati (2011:6), menjelaskan bahwa *UML* singkatan dari *Unified Modeling Language* yang berarti bahasa permodelan standar.

Munawar (2005:19), menjelaskan bahwa *UML (Unified Modelling Language)* adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi obyek.

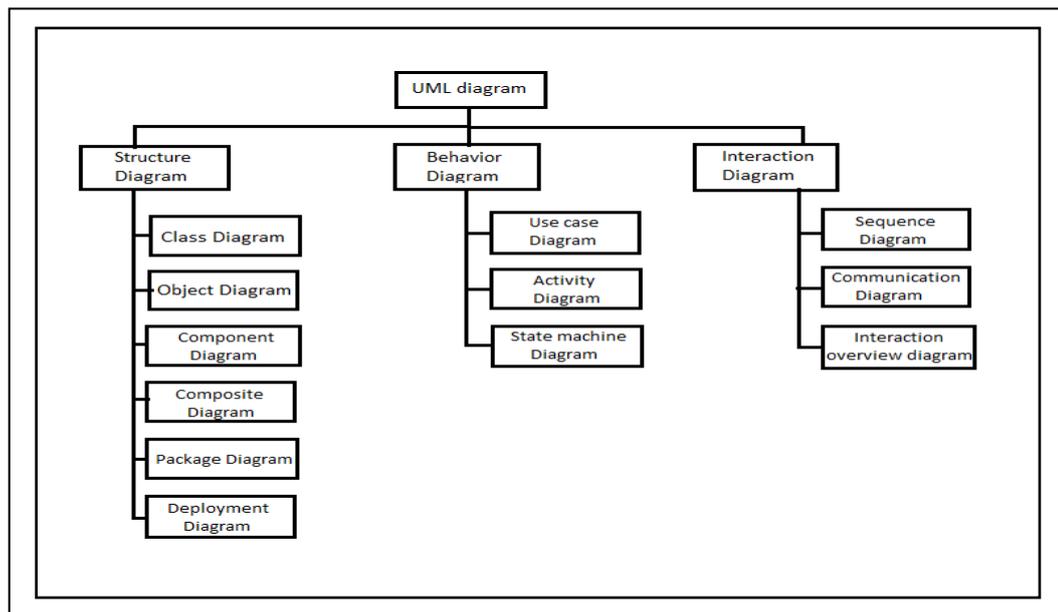
UML menyediakan bahasa pemodelan visual yaitu proses penggambaran informasi-informasi secara grafis dengan notasi-notasi baku yang telah disepakati sebelumnya. Dengan menggunakan pemodelan *UML* ini, pengembang dapat melakukan:

- a. tinjauan umum bagaimana arsitektur sistem secara keseluruhan.
- b. Penelaahan bagaimana objek-objek dalam sistem saling mengirimkan pesan (*message*) dan saling bekerjasama satu sama lain.
- c. Menguji apakah sistem/perangkat lunak sudah berfungsi seperti yang seharusnya.

Dokumentasi sistem/perangkat lunak untuk keperluan-keperluan tertentu di masa yang akan datang.

2.2.4.1. Macam-macam Diagram UML

Pada *UML* terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar dibawah ini:



(Sumber: Sukamto dan Shalahuddin 2013:140)

Gambar 2.2 Macam-macam Diagram UML

Berikut ini penjelasan singkat dari pembagian kategori tersebut.

a. Structure Diagram

Yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.

b. Behavior Diagram

Yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.

c. Interaction Diagram

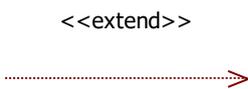
Yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem.

1. Usecase Diagram

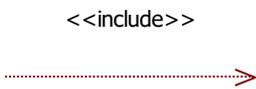
Munawar (2005:63), menjelaskan bahwa *Usecase Diagram* adalah deskripsi fungsi dari sebuah *system* dari perspektif pengguna.

Sukamto dan Shalahuddin (2013:155), menjelaskan bahwa *Usecase* atau diagram *usecase* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat.

Tabel 2.1 Simbol-simbol *Usecase Diagram*

No	Simbol	Deskripsi
1.	Usecase 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.
2.	Aktor / <i>actor</i> 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun symbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
3.	Asosiasi / <i>association</i> 	Komunitas antara aktor dan usecase yang berpartisipasi pada usecase atau usecase memiliki interaksi dengan aktor.
4.	Ekstensi / <i>extend</i> 	Relasi <i>usecase</i> tambahan ke sebuah <i>usecase</i> dimana usecase yang ditambahkan dapat berdiri sendiri walau tanpa <i>usecase</i> tamhanan itu; biasanya <i>usecase</i> yang menjadi <i>extend</i> -nya merupakan jenis yang sama dengan <i>usecase</i> yang menjadi induknya.
5.	Generalisasi / <i>Generalization</i> 	Hubungan generalisasi dan spesifikasi (umum-khusus) antara dua buah usecase dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.

Lanjutan tabel 2.1 Simbol-simbol *Usecase Diagram*

6.	<i>Include</i> 	Relasi usecase tambahan ke sebuah <i>usecase</i> di mana <i>usecase</i> yang ditambahkan memerlukan <i>usecase</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>usecase</i> ini; <i>include</i> berarti <i>usecase</i> yang ditambahkan akan selalu dipanggil saat <i>usecase</i> tambahan dijalankan.
----	---	---

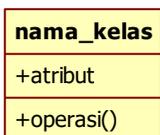
(Sumber: Sukanto dan Shalahuddin,2013:155)

2. Class Diagram

Widodo dan Herlawati (2011:3), menjelaskan bahwa *class diagram* adalah penggambaran satu set objek yang memiliki atribut dan *behavior* yang sama.

Sukanto dan Shalahuddin (2013:141), menjelaskan bahwa diagram kelas atau class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem.

Tabel 2.2 Simbol-simbol *Class Diagram*

No	Simbol	Deskripsi
1.	Kelas 	Kelas pada struktur sistem.
2.	Antar muka/ <i>interface</i> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
3.	Asosiasi / <i>association</i> 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .



Lanjutan Tabel 2.2 Simbol-simbol *Class Diagram*

4.	Asosiasi berarah / <i>directed association</i> 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
5.	Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus).
6.	Kebergantungan / <i>dependency</i> 	Relasi antar kelas dengan makna kebergantungan antar kelas.
	Agregasi / <i>aggregation</i> 	Relasi antar kelas dengan makna semua-bagian (whole-part).

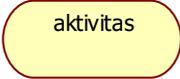
(Sumber: Sukanto dan Shalahuddin, 2013:146)

3. Activity Diagram

Munawar (2005:109), menjelaskan bahwa *activity diagram* adalah teknik untuk mendeskripsikan logika prosedural, proses bisnis dan aliran kerja dalam banyak kasus.

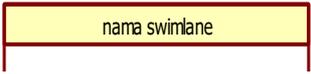
Sukanto dan Shalahuddin (2013:161), menjelaskan bahwa diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak.

Tabel 2.3 Simbol-simbol *Activity Diagram*

No	Simbol	Deskripsi
1.	Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.



Lanjutan Tabel 2.3 Simbol-simbol *Activity Diagram*

3.	Percabangan / <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4.	Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5.	Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6.	Swimlane  atau 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

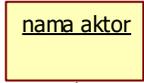
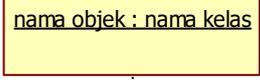
(Sumber: Sukamto dan Shalahuddin,2013:162)

4. *Sequence Diagram*

Munawar (2005:87), *sequence diagram* digunakan untuk menggambarkan perilaku pada sebuah skenario. Diagram ini menunjukkan sejumlah contoh obyek dan message (pesan) yang diletakkan diantara obyek-obyek ini di dalam usecase.

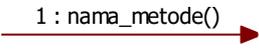
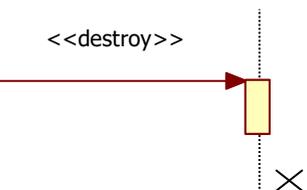
Sukamto dan Shalahuddin (2013:165), diagram sekuensial menggambarkan kelakuan objek pada *usecase* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek.

Tabel 2.4 Simbol-simbol *Sequence Diagram*

No	Simbol	Deskripsi
1.	<p>Aktor</p>  <p>nama aktor</p> <p>atau</p>  <p>nama aktor</p> <p>tanpa waktu aktif</p>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
2.	<p>Garis hidup / <i>lifeline</i></p> 	menyatakan kehidupan suatu objek.
3.	<p>Objek</p>  <p>nama objek : nama kelas</p>	Menyatakan objek yang berinteraksi pesan.
4.	<p>Waktu aktif</p> 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya.
5.	<p>Pesan tipe create</p>  <p><<create>></p>	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.



Lanjutan Tabel 2.4 Simbol-simbol *Sequence Diagram*

6.	Pesan tipe call 	Menyatakan suatu objek memanggil operasi/ metode yang ada pada objek lain atau dirinya sendiri, arah panah mengarah pada objek yang memiliki operasi/ metode, karena ini memanggil operasi/ metode maka operasi/ metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.
7.	Pesan tipe send 	Menyatakan bahwa suatu objek mengirimkan data/ masukan/ informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
8.	Pesan tipe return 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.
9.	Pesan tipe destroy 	Menyatakan suatu objek mengakhiri hidup objek lain, arah panah mengarah pada objek yang diakhiri, sebaliknya jika ada create maka ada destroy.

(Sumber: Sukamto dan Shalahuddin,2013:165)

5. Collaboration Diagram

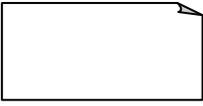
Munawar (2005:101), menjelaskan bahwa *collaboration diagram* adalah diagram yang menunjukkan *message-message* obyek yang dikirimkan satu sama lain.

Seperti *sequence diagram*, *collaboration diagram* juga digunakan untuk memperlihatkan aliran-aliran pada *usecase*. Sementara *sequence diagram*



berurutan menurut waktu, *collaboration diagram* berfokus pada relasi-relasi yang terjadi antara objek yang satu dengan objek-objek yang lainnya. Nugroho (2005:93). Diagram kolaborasi adalah diagram interaksi yang menekankan organisasi struktural dari objek-objek yang menerima serta mengirim pesan (*message*). Nugroho (2005:19).

Tabel 2.5 Simbol-simbol *Collaboration Diagram*

No.	Simbol	Keterangan
1		<i>Note</i>
2		<i>Class</i>
3		<i>Interface</i>
4		<i>Association</i>
5		<i>Dependency & Realize</i>



2.3. Teori Program

2.3.1. Pengertian Pemrograman *PHP*

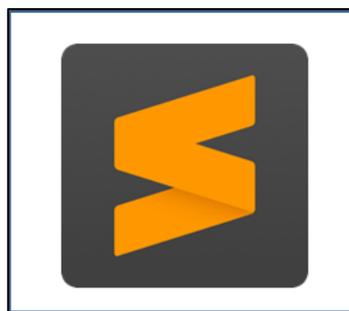


Gambar 2.1. Logo PHP

PHP (*HyperText Preprocessor*) adalah sebuah bahasa utama *script server side* yang disisipkan pada HTML yang dijalankan di *server*, dan juga bisa digunakan untuk membuat aplikasi desktop.

Menurut Betha Sidik, dalam bukunya yang berjudul Pemrograman Web Dengan *PHP* (2012 : 4), menyebutkan bahwa *PHP* merupakan secara umum dikenal dengan sebagai bahasa pemrograman *script-script* yang membuat dokumen HTML secara *on the fly* yang dieksekusi di server web, dokumen HTML yang dihasilkan dari suatu aplikasi bukan dokumen HTML yang dibuat dengan menggunakan editor teks atau editor HTML, dikenal juga sebagai bahasa pemrograman *server side*.

2.3.2. Sublime Text



Gambar 2.2. Logo Sublime Text

Sublime Text adalah aplikasi editor untuk kode dan teks yang dapat berjalan diberbagai platform *operating system* dengan menggunakan teknologi Python API. Terciptanya aplikasi ini terinspirasi dari aplikasi Vim, aplikasi ini sangatlah



fleksibel dan *powerfull*. Fungsionalitas dari aplikasi ini dapat dikembangkan dengan menggunakan *sublime-packages*.

Sublime Text bukanlah aplikasi *open source* dan juga aplikasi yang dapat digunakan dan didapatkan secara gratis, akan tetapi beberapa fitur pengembangan fungsionalitas (*packages*) dari aplikasi ini merupakan hasil dari temuan dan mendapat dukungan penuh dari komunitas serta memiliki linsensi aplikasi gratis. Sublime Text mendukung berbagai bahasa pemrograman dan mampu menyajikan fitur *syntax highlight* hampir di semua bahasa pemrogramman yang didukung ataupun dikembangkan oleh komunitas seperti; *C, C++, C#, CSS, D, Dylan, Erlang, HTML, Groovy, Haskell, Java, JavaScript, LaTeX, Lisp, Lua, Markdown, MATLAB, OCaml, Perl, PHP, Python, R, Ruby, SQL, TCL, Textile* dan *XML*. Biasanya bagi bahasa pemrograman yang didukung ataupun belum terdukung secara *default* dapat lebih dimaksimalkan atau didukung dengan menggunakan *add-ons* yang bisa *download* sesuai kebutuhan *user*.

2.3.3. Pengertian MySQL

Shalahuddin (2013:46), menjelaskan bahwa SQL (*Structured Query Language*) adalah bahasa yang digunakan untuk mengelola data pada RDBMS. SQL awalnya dikembangkan berdasarkan teori aljabar relasional dan kalkulus.

Prasetio (2014: 29), menjelaskan bahwa *MySQL* adalah sebuah database, dan dapat anda bayangkan seperti sebuah tempat penyimpanan.

Sibero (2014:97), menjelaskan bahwa *MySQL* atau dibaca 'My Sekuel' dengan adalah suatu *RDBMS (Relation Database Management System)* yaitu aplikasi sistem yang menjelaskan fungsi pengolahan data. Berikut tipe-tipe data pada *MySQL*:

Tabel 2.10. Tipe Data pada *MySQL*

No	Tipe Data	Ukuran (Bytes)
1.	TINYINT	1
2.	SMALLINT	2
3.	MEDIUM	3



Lanjutan Tabel 2.10 Tipe Data pada *MySQL*

4.	INT	4
5.	BIGINT	8
6.	FLOAT(p)	4
7.	FLOAT	4
8.	DOUBLE	8
9.	REAL	8
10.	DECIMAL (L,D)	L + 1
11.	NUMERIC (L,D)	L + 1
12.	BIT (L)	8
13.	DATE	3
14.	TIME	3
15.	DATETIME	8
16.	TIMESTAMP	4
17.	YEAR	1
18.	CHAR (L)	L
19.	BINARY(L)	L + 1
20.	VARCHAR (L)	L + 1
21.	VARBINARY (L)	L + 1
22.	TINYBLOB	L + 1
23.	TINYTEXT	L + 1
24.	BLOB	L + 2
25.	TEXT	L + 2
26.	MEDIUMBLOB	L + 3
27.	MEDIUMTEXT	L + 3
28.	LOB	L + 4
29.	LONGTEXT	L + 4
30.	ENUM ('nilai',...)	1 – 2
31.	SET ('nilai',...)	1 – 8



2.4. Referensi Penelitian Terkait

Terdapat penelitian yang memiliki keterkaitan dengan topik peninjauan kinerja menggunakan Dashboard. Penelitian tersebut masih 5 (lima) tahun terakhir yang selanjutnya digunakan sebagai rujukan. Berikut ini beberapa penelitian terkait.

Dalam jurnal yang ditulis oleh Ilhamsyah dan Agus Rahmayudha dengan judul **“Perancangan Model *Dashboard* Untuk Monitoring Evaluasi Mahasiswa”** pada tahun 2017. Dalam penelitian ini disebutkan bahwa tidak adanya sistem yang pas untuk monitoring dan evaluasi mahasiswa di bidang pendidikan di Jurusan Sistem Informasi FMIPA Universitas Tanjungpura Pontianak membuat *dashboard* digunakan sebagai sistem monitoring evaluasi mahasiswa di bidang pendidikan di Jurusan Sistem Informasi FMIPA Universitas Tanjungpura Pontianak sebagai sarana untuk mengukur kualitas mahasiswa serta mengambil keputusan dalam menentukan strategi kedepan pembinaan mahasiswa di lingkungan Jurusan Sistem Informasi FMIPA Untan Pontianak.

Menurut penelitian Agus Prasetyo Utomo, S.Kom, M.M, M.Kom dan Irfan Ismail Sungkar, S.Kom, M.Kom pada tahun 2014 dalam jurnal yang berjudul **“Analisis dan Perancangan Dashboard untuk Monitoring dan Evaluasi Pasien Rawat Inap”** disebutkan bahwa pencatatan masih manual dari setiap kunjungan perawat sehingga masih lama dan tidak efektif. Ketika dokter ingin mengecek perkembangan kesehatan pasien atau riwayat pasien mengalami kesulitan karena informasi yang diberikan tidak mudah untuk dipahami. Maka dirancanglah “alat” dengan metode *Pureshare*, sehingga dapat digunakan dalam pengambilan keputusan untuk memantau dan mengevaluasi kinerja rumah sakit (dalam hal ini *monitoring resume* pasien rawat inap) dengan memanfaatkan digital informasi yang terintegrasi dengan *e-medical* yang ditampilkan dalam bentuk *dashboard*.

Dari penelitian yang ditulis oleh Sholeh dkk pada tahun 2013 mengenai **“Metode Peninjauan Dashboard dari *Business Intelligence* Untuk Membuat Keputusan Lebih Baik”** disebutkan bahwa secara umum pengambilan keputusan suatu organisasi masih menggunakan grafik dan tabel sehingga tidak



mengefektifkan kinerja suatu organisasi dikarenakan terlambatnya proses pengambilan keputusan. Oleh karena itu, *dashboard* merupakan solusi yang tepat dalam penyajian informasi dari proses *business intelligence* yaitu memberikan tampilan antarmuka dengan berbagai bentuk seperti diagram, laporan, *visual indicator*, mekanisme *alert*, yang dipadukan dengan informasi yang dinamis dan relevan.

Menurut penelitian yang ditulis oleh Agus Prasetyo Utomo, Hari Murti, dan Rara Sri Artati Rejeki dalam jurnal berjudul “**Sistem Monitoring dan Evaluasi Kinerja Program Studi dengan Metode Performance Dashboard**” pada tahun 2013. Dalam penelitian tersebut dijelaskan bahwa belum adanya sistem pengukuran kinerja program studi dalam bentuk dashboard yang ditujukan secara khusus sesuai dengan standar Badan Akreditasi Nasional Perguruan Tinggi (BAN-PT). Sehingga untuk melakukan monitoring dan evaluasi kinerja program studi secara tidak langsung masih dilakukan secara manual oleh semua pihak yang berkepentingan terhadap kinerja program studi di lingkungan universitas.

Sedangkan dalam jurnal “**Analisis Dan Desain Dasbor Inteligensia Bisnis Untuk Memantau Kinerja Organisasi di KPPN Surabaya I**” yang ditulis oleh Andi Saputra, Rully Agus Hendrawan dan Muhammad Priandi pada tahun 2013. Dalam penelitian ini disebutkan bahwa pemantauan kinerja organisasi di KPPN masih dilakukan selama 3 bulan sekali, hal ini berakibat KPPN kurang responsif terhadap tantangan dan kesempatan yang ada sehingga menghambat pencapaian kinerja organisasi. Berdasarkan masalah ini diperlukan sebuah aplikasi pemantau kinerja organisasi yang dapat dilihat setiap saat dengan data terbaru. Maka dibuatlah dasbor menggunakan metode Pureshare 6 dengan pendekatan *top-down design*. Hasil dari penelitian ini adalah sebuah dashboard dengan tipe taktikal, yang dapat digunakan oleh Kepala Kantor dan Pejabat KPPN.