



BAB II

TINJAUAN PUSTAKA

2.1 Teori Judul

2.1.1 Pengertian Aplikasi

Menurut Solichin (2016:1), aplikasi atau perangkat lunak (*software*) merupakan bagian yang tidak terpisahkan dari suatu sistem komputer, disamping keberadaan pengguna (*brainware*), perangkat keras (*hardware*) dan jaringan (*networking*).

Sedangkan menurut Kadir (2017:3), istilah program dan aplikasi lebih sering untuk menyatakan perangkat lunak. Dikalangan profesional teknologi informasi, istilah program biasa digunakan untuk hasil karya mereka yang berupa instruksi- instruksi untuk mengendalikan computer. Disisi pemakai, hal seperti itu biasa disebut aplikasi.

Dari pengertian diatas dapat disimpulkan bahwa aplikasi merupakan suatu perangkat lunak, software yang bertugas untuk melakukan sekumpulan perintah atau tugas-tugas tertentu yang dijalankan oleh user yang saling berkaitan

2.1.2 Pengertian Evaluasi

Menurut Lukum (2015:28), evaluasi dapat didefinisikan sebagai penilaian pencapaian tujuan melalui pengumpulan dan analisis data yang berguna untuk membuat keputusan dari suatu program.

Menurut Mariko, dkk (2016:56), evaluasi merupakan suatu proses yang mencakup pengukuran dan mungkin juga testing, yang juga berisi pengambilan keputusan tentang nilai.

Dari pengetahuan diatas dapat disimpulkan bahwa evaluasi merupakan suatu proses sistematis dalam memeriksa, menentukan, membuat keputusan atau menyediakan informasi terhadap pencapaian suatu program.



2.1.3 Pengertian Kinerja

Menurut Abas (2017:21), kinerja atau prestasi kerja (*performance*) diartikan sebagai ungkapan kemampuan yang didasari oleh pengetahuan, sikap, keterampilan dan motivasi dalam menghasilkan sesuatu.

Menurut Galton dan Simon yang dikutip oleh Abas (2017:23), kinerja adalah aktivitas seseorang dalam melakukan tugas pokok yang dibebankan kepadanya.

Berdasarkan pengertian diatas, dapat disimpulkan bahwa kinerja adalah hasil kerja secara kualitas dan kuantitas yang dapat dicapai oleh seorang pegawai dalam melaksanakan tugas sesuai dengan tanggung jawab yang diberikan kepadanya.

2.1.4 Pengertian Dosen

Menurut Toatubun dan Rijal (2018:30), dosen adalah tenaga pendidik pada perguruan tinggi yang khusus diangkat dengan tugas utama mengajar.

Menurut Undang– Undang Guru dan Dosen nomor 14 tahun 2005, dosen adalah pendidik professional dari ilmuwan dengan tugas utama metnransformasikan, mengembangkan dan menyebarluaskan ilmu pengetahuan, teknologi dan seni melalui pendidikan”.

Berdasarkan pengertian diatas, dapat disimpulkan bahwa dosen adalah tenaga pengajar pada perguruan tinggi yang ikut berperan dalam usaha pembentukan sumber daya manusia yang potensial di bidang pembangunan.

2.1.5 Pengertian Jurusan Bahasa Inggris

Jurusan bahasa Inggris adalah salah satu jurusan yang berada di Politeknik Negeri Sriwijaya. Jurusan Bahasa Inggris berdiri pada tahun akademik 2004/2005 dengan Program Studi Bahasa Inggris Bisnis Pariwisata dan Perhotelan berdasarkan izin dari Dirjen Dikti nomor 3818/D/T/2003.

2.1.6 Pengertian Web

Menurut Destiningrum dan Adrian (2017:32), web adalah sebuah software yang berfungsi untuk menampilkan dokumen-dokumen pada suatu web yang



membuat pengguna dapat mengakses internet melalui software yang terkoneksi dengan internet.”.

Menurut Solichin (2016:3), web merupakan bagian dari layanan yang dapat berjalan di atas teknologi internet.

Berdasarkan pengertian diatas, dapat disimpulkan bahwa *web* adalah koleksi sumber informasi kaya grafis yang saling berhubungan satu sama lain dalam internet yang lebih besar.

2.1.7 Pengertian Politeknik Negeri Sriwijaya

Politeknik Negeri Sriwijaya adalah salah satu Politeknik Negeri di Indonesia dan berpusat di Kota Palembang yang secara resmi dibuka pada tanggal 20 September 1982. Sekarang Politeknik Negeri Sriwijaya sudah memiliki 9 Jurusan, yaitu Teknik Sipil, Teknik Mesin, Teknik Elektro, Teknik Kimia, Teknik Komputer, Akuntansi, Administrasi Bisnis, Manajemen Informatika, dan Bahasa Inggris.

2.1.8 Pengertian Aplikasi Evaluasi Kinerja Dosen pada Jurusan Bahasa Inggris Berbasis *Web* di Politeknik Negeri Sriwijaya

Aplikasi evaluasi kinerja dosen pada Jurusan Bahasa Inggris Berbasis *Web* di Politeknik Negeri Sriwijaya adalah aplikasi yang dibuat untuk mempermudah Ketua Jurusan memantau dosen yang mengajar di Jurusan Bahasa Inggris Politeknik Negeri Sriwijaya dan mempermudah untuk penyimpanan data secara *online*.

2.2 Teori Khusus

2.2.1 *Object Oriented Program (OOP)*

Menurut Abdullah (2017:1), OOP (*Object Oriented Program*) merupakan teknik pemrograman dengan menggunakan konsep objek. Tujuan dari OOP adalah untuk memudahkan programmer dalam pembuatan program dengan menggunakan konsep objek yang ada dalam kehidupan sehari-hari. Jadi setiap bagian



permasalahan adalah objek, dan objek itu sendiri merupakan gabungan dari beberapa objek yang lebih kecil.

Sebuah objek pada OOP memiliki data atau disebut property yang menjelaskan tentang sifat-sifat objek tersebut. Seperti sebuah handphone dapat memiliki data warna, merk, ukuran layar, dan sebagainya. Begitu juga dengan objek-objek yang ada didalamnya seperti layar memiliki data berupa lebar, tinggi dan sebagainya.

Selain memiliki property, sebuah objek dalam OOP memiliki method berupa fungsi yang dapat dipanggil untuk melakukan tindakan atau merubah nilai dari property yang ada di dalamnya. Seperti handphone dapat melakukan tindakan merekam, restart, memanggil, mengirim pesan dan sebagainya. Handphone juga dapat diganti *casing* untuk mengubah warnanya (mengubah nilai property).

2.2.2 *Unified Modelling Language (UML)*

Menurut Rosa A.S dan M. Salahuddin (2018:13), *Unified Modeling Language (UML)* adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. UML hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek.

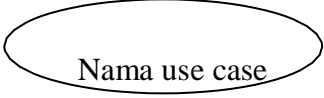
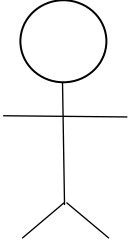

2.2.3 *Use Case Diagram*

Rosa A.S dan M. Salahuddin (2018:155), *use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Berikut adalah simbol-

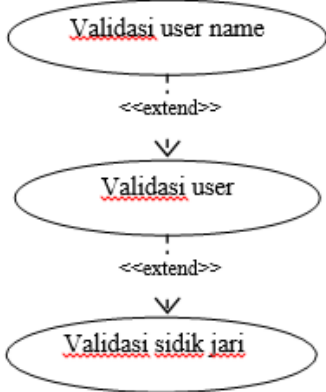
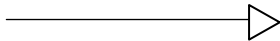
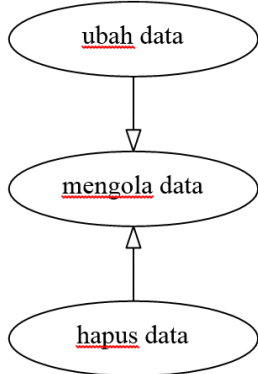


simbol yang ada pada diagram *use case* : Diagram kelas dibuat agar pembuat program atau *programmer* membuat kelas-kelas sesuai rancangan di dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron.

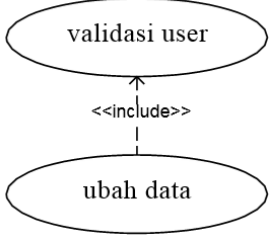
Tabel 2.1 Simbol-simbol *Use Case* diagram

| No. | Simbol | Deskripsi |
|-----|---|---|
| 1. | <p><i>Use case</i></p>  | Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama <i>use case</i> . |
| 2. | <p>Aktor/<i>actor</i></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. |
| 3. | <p>Assosiasi/<i>association</i></p>  | Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor. |
| 4. | <p>Exstensi/<i>extend</i></p> <p><<extend>></p> | Relasi <i>use case</i> tambahan kesebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu, mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi |



| | | |
|----|---|---|
| | | <p>objek, biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, misal :</p>  <p>Arah panah mengarah pada <i>use case</i> yang ditambahkan, biasanya <i>use case</i> yang menjadi <i>extend</i>-nya merupakan jenis yang sama dengan <i>use case</i> yang menjadi induknya.</p> |
| 5. | <p>Generalisasi/<i>generalization</i></p>  | <p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua <i>buah use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya :</p>  |

Lanjutan Tabel 2.1 Simbol-simbol *Use Case* diagram



| No. | Simbol | Deskripsi |
|-----|--------|---|
| | |  <p>Kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.</p> |

Sumber : Rosa A.S dan M. Salahuddin (2018:156-158)




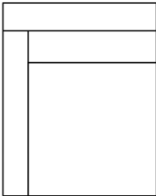
2.2.4 Activity Diagram

Rosa A.S dan M. Salahuddin (2018:161), diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Berikut adalah simbol-simbol yang ada pada diagram aktivitas :

Tabel 2.2 Simbol-simbol *activity diagram*

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

Lanjutan Tabel 2.2 Simbol-simbol *activity diagram*

| No. | Simbol | Deskripsi |
|-----|---|---|
| 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 oleh sistem, sebuah diagram aktivitas memiliki sebuah status akhir. |
| 6. | Swimlane  | Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi. |

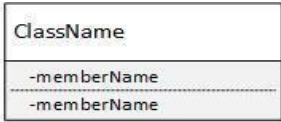


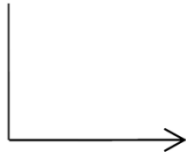

Sumber : Rosa A.S dan M. Salahuddin (2018:162-163)

2.2.5 Class Diagram


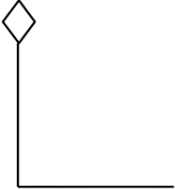
Rosa A.S dan M. Salahuddin (2018:141), diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan *method* atau operasi. Berikut penjelasan atribut dan *method* :

1. Atribut merupakan *variable*-variabel yang dimiliki oleh suatu kelas.
2. Operasi atau *method* adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Tabel 2.3 Simbol-simbol *class diagram*

| No. | Simbol | Deskripsi |
|-----|--|--|
| 1. | Kelas  | Kelas pada struktur sistem |
| 2. | Antarmuka/ <i>interface</i>  | Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek |
| 3. | Asosiasi/ <i>association</i>  | Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> |
| 4. | Asosiasi berarah/ <i>directed association</i>  | Relasi antar kelas 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) |

Lanjutan Tabel 2.3 Simbol-simbol *class diagram*

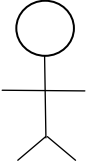
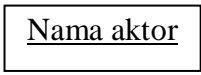

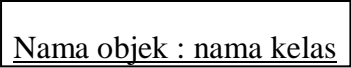

| No. | Simbol | Deskripsi |
|-----|--|--|
| 6. | Kebergantungan/ <i>dependensi</i>  | Relasi antar kelas dengan makna kebergantungan antar kelas |
| 7. | Agregasi/ <i>aggregation</i>  | Relasi antar kelas dengan makna semua-bagian (<i>whole-part</i>) |

Sumber : Rosa A.S dan M. Salahuddin (2018:146-147)

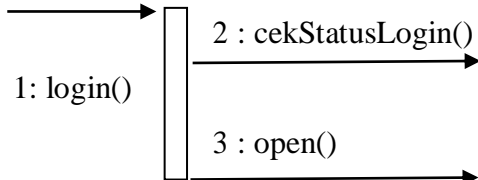
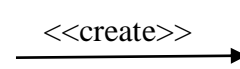
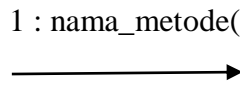
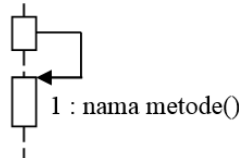
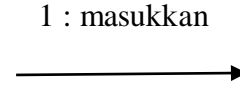
2.2.6 Sequence Diagram

Rosa A.S dan M. Salahuddin (2018:165), diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dengan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*. Banyaknya diagram sekuen yang harus digambar adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksinya pesan sudah dicakup dalam diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak. Berikut adalah simbol-simbol yang ada pada diagram sekuen :

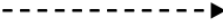
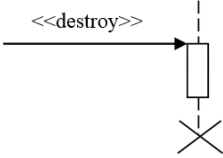
Tabel 2.4. Simbol-simbol *sequence diagram*

| No. | Simbol | Deskripsi |
|-----|---|--|
| 1. | <p>Aktor</p>  <p>Atau</p>  <p>Tanpa waktu aktif</p> | <p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan dalam menggunakan kata benda diawal frase nama aktor.</p> |
| 2. | <p>Garis hidup/<i>lifeline</i></p>  | <p>Menyatakan kehidupan suatu objek</p> |
| 3. | <p>Objek</p>  | <p>Menyatakan objek yang berinteraksi pesan</p> |
| 4. | <p>Waktu aktif</p>  | <p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semuanya yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya</p> |

Lanjutan Tabel 2.4. Simbol-simbol *sequence diagram*

| No. | Simbol | Deskripsi |
|-----|--|--|
| | |  <p>Maka cekStatusLogin() dan open() dilakukan didalam metode login(). Aktor tidak memiliki waktu aktif</p> |
| 5. | Pesan tipe <i>create</i>  | Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat |
| 6. | Pesan tipe <i>call</i>  | Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri,  <p>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</p> |
| 7. | Pesan tipe <i>send</i>  | Menyatakan bahwa suatu objek mengirimkan data/masukkan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim. |

Lanjutan Tabel 2.4. Simbol-simbol *sequence diagram*

| No. | Simbol | Deskripsi |
|-----|---|---|
| 8. | Pesan tipe <i>return</i> 1 : keluaran  | 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 <i>destroy</i>  | Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaliknya jika ada <i>create</i> maka ada <i>destroy</i> |

Sumber : Rosa A.S dan M. Salahuddin (2018:165-167)

2.3 Penelitian Terdahulu

Berdasarkan penelitian yang telah dilakukan oleh Irmawati Carolina dan Adi Supriyatna dengan judul Penerapan Metode Pengembangan *Extreme Programming* Dalam Perancangan Aplikasi penghitungan Kuota SKS Mengajar Dosen pada tahun 2019, ISSN: 2580-4316. Merancang suatu aplikasi tentunya membutuhkan suatu metode yang khusus. Metode yang digunakan pada perancangan aplikasi ini adalah *Extreme Programming* yang merupakan metode yang memiliki empat tahapan dalam pelaksanaannya, yaitu perencanaan, perancangan, pengkodean dan pengujian. Hasil dari penelitian ini berupa aplikasi pemetaan kuota sks mengajar dosen yang dapat memberikan kemudahan dan keakuratan dalam perhitungan jumlah sks mengajar dosen dengan cepat dan tepat, mampu mengurangi kesalahan dalam proses perhitungan dan dapat meminimalisir komplain dari dosen terkait honor mengajar yang diperoleh.

Menurut penelitian yang dibuat oleh Adi Supriyatna pada tahun 2018 yang berjudul Metode *Extreme Programming* pada pembangunan web aplikasi seleksi



peserta pelatihan kerja p-ISSN: 1979-9160, e-ISSN : 2549-7901, metode *extreme programming* merupakan sebuah proses rekayasa perangkat lunak yang cenderung menggunakan pendekatan berorientasi objek. Tujuan dari pembuatan aplikasi ini untuk memberikan kemudahan bagi calon peserta seleksi pelatihan kerja untuk melakukan pendaftaran, ujian dan mendapatkan informasi mengenai periode penerimaan peserta pelatihan kerja secara online.

Dalam jurnal penelitian yang dibuat oleh Muhathir pada tahun 2015 yang berjudul Penentuan Kelayakan Penerbitan Paper Menggunakan Fuzzy Logic (Studi Kasus Universitas Malikussaleh ISSN : 2470-8735, dalam menentukan layak paper tersebut terbit berdasarkan seluruh nilai yang ada dengan cepat dan tepat. Penggunaan logika fuzzy dengan aplikasi matlab 7.9 diyakini dapat memberikan solusi dari permasalahan yang ada sehingga didapatkan hasil layak tidaknya paper tersebut di publikasikan oleh universitas malikussaleh.

Menurut penelitian yang dibuat oleh Halimahtus Mukminna,dkk pada tahun 2017 yang berjudul Simulasi Kinerja Siswa Dengan Metode *Fuzzy Inference Sugeno* Menggunakan Aplikasi *Matlab*, dalam pertimbangan evaluasi kinerja siswa ini menggunakan logika fuzzy dengan metode *inference system sugeno*. Hasil yang ditampilkan dengan perhitungan manual dan perhitungan Matlab sebagai pembandingnya hasil perhitungan secara manual nilai result 45,5 sedangkan pada perhitungan matlab nilai result sebesar 48,5. Sehingga dapat disimpulkan selisih yang disebabkan tingkat akurasi hasil *inference rule* pada perhitungan manual kurang efektif bahkan terkadang banyak *inference rule* yang harus disesuaikan.

Berdasarkan penelitian yang telah dilakukan oleh Febri Maspiyanti dan Nadya Hadiyanti yang berjudul Robot Pemadam Api Menggunakan Fuzzy Logic pada tahun 2017, penerapan *Fuzzy Logic* pada robot pemadam api adalah sebagai kendali kecepatan Motor DC robot dengan masukan jarak. Penelitian ini menghasilkan tingkat akurasi sebesar 92,63% untuk berjalan lurus, jalan berbelok 82,82%, pembacaan jarak oleh Sensor Ultrasonik 81,59%, deteksi api dengan Flame Sensor 87,25%, dan tingkat akurasi memadamkan api dengan L9110 Fan Module sebesar 52,11%.