



BAB II
TINJAUAN PUSTAKA
DAN LANDASAN TEORI

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Beberapa penelitian yang relevan dilakukan peneliti sebagai berikut: Penelitian Sistem Pendukung Keputusan dengan Metode *Weighted Product* (WP) sebelumnya pernah dilakukan oleh beberapa peneliti, salah satunya oleh tim pengembang yang bernama Dona, Kiki Yasdomi, dan Urfi Utami program studi sistem informasi mahasiswa Universitas Pasir Pengaraian. Penelitian ini pada tahun 2018 membuat “Sistem Pendukung Keputusan Penentuan Karyawan Terbaik dengan Metode *Weighted Product* (WP) (Studi Kasus: Universitas Pasir Pengaraian).” menggunakan 5 kriteria pemilihan diantaranya kedisiplinan, kinerja, tanggung jawab, prestasi, dan kerjasama. Sistem ini dikembangkan menggunakan metode analisis dengan pendekatan terstruktur (*Structured Approach*) yang lengkap dengan alat dan teknik yang dibutuhkan dalam sistem. Dapat disimpulkan bahwa dari hasil penelitian, sistem pendukung keputusan menggunakan Metode *Weighted Product* digunakan untuk memilih karyawan terbaik yang diharapkan memacu semangat kerja karyawan dalam meningkatkan operasional demi kemajuan dan perkembangan perguruan tinggi[5].

Penelitian selanjutnya mengenai sistem pendukung keputusan metode *Weighted Product* (WP) pernah diteliti juga oleh Dwi Cahyanto Yoni dan Hindayati Mustafidah mahasiswa Universitas Muhammadiyah Purwokerto dengan judul “Penerapan Metode *Weighted Product* untuk Pemilihan Mahasiswa Lulusan Terbaik di Fakultas Teknik Universitas Muhammadiyah Purwokerto (UMP)” pada tahun 2016. Sistem yang dibuat diharapkan memberi kemudahan bagi pihak Fakultas Teknik UMP dalam mengelola nilai dan data mahasiswa sehingga mampu menentukan mahasiswa lulusan terbaik berdasarkan kriteria IPK, masa studi, nilai C maksimal 1 dan tidak ada nilai D. Hal tersebut karena proses pemilihan mahasiswa lulusan terbaik di Fakultas Teknik ini masih menggunakan cara manual dan belum memakai sistem. Proses input dan perhitungan masih melihat dari data transkrip nilai mahasiswa. Sistem ini menggunakan model *waterfall* sebagai metode pengembangannya. Perancangan sistem digambarkan menggunakan

flowchart dan *use case diagram*. *Flowchart* dalam perancangan sistem ini yaitu *flowchart* pengembangan sistem pendukung keputusan dan *flowchart* aplikasi sistem. *Use case diagram* untuk menerangkan proses-proses yang dilakukan user dalam menjalankan sistem[6].

Selain itu penelitian sistem pendukung keputusan Himpunan Mahasiswa Jurusan (HMJ) pernah diteliti oleh Diana Corry, Yuliani Indrianingsih dari Sekolah Tinggi Teknologi Adisutjipto Yogyakarta pada tahun 2014 dengan judul “Sistem Pendukung Keputusan Penerimaan Anggota Himpunan Mahasiswa Jurusan Teknik Informatika dengan menggunakan Metode Electre” untuk mengatasi masalah penerimaan pengurus HMJ, pengembang membuat sistem tersebut dengan tujuan mempermudah dalam penerimaan pengurus HMJ untuk diterapkan atau digunakan oleh Sekolah Tinggi Teknologi Adisutjipto Yogyakarta sesuai dengan kriteria kebutuhan. Dalam penerimaan pengurus HMJ, diperlukan beberapa kriteria penilaian yang didapatkan dari beberapa tes, kriteria tersebut adalah minat, keaktifan, kepercayaan diri, psikologi, dan IPK[7].

Sistem pendukung keputusan pernah diteliti juga oleh Reynold Adiputra dan Bagas Mulyawan pada tahun 2019 yang berjudul “Pembuatan Program Aplikasi Sistem Pendukung Keputusan Pemilihan Vendor ERP pada PT Sinar Jaya Abadi dengan Menggunakan Metode *Weighted Product*”. Tujuan dibangunnya sistem ini guna mempermudah perusahaan dalam melakukan pemilihan vendor ERD. Dibuatnya sistem ini karena dalam proses bisnisnya selama ini perusahaan masih menggunakan Microsoft Excel dalam mencatat semua transaksi di semua bagian. Penggunaan Excel dianggap dan dirasakan oleh manajemen tidak praktis, efektif dan lama karena file-file Excel tidak terintegrasi antar satu dengan lainnya antar bagian. Maka diperlukan sistem terintegrasi agar dapat mengatasi permasalahan PT. Sinar Jaya Abadi. Pengembangan sistem ini menggunakan metode pendekatan terstruktur untuk memberikan gambaran secara umum mengenai sistem yang ada saat ini. Rancang sistem ini menggunakan *Entity Relationship Diagram* (ERD) dan hubungan antar tabel. Adapun dalam pemilihan vendor ERP diperlukan kriteria-kriteria seperti harga, *hardware support*, garansi, User License dan lama implementasi[8].

Pada tahun 2019 sistem pendukung keputusan juga pernah diteliti oleh Roni, Sumijan, Julius Santony dengan judul “Metode *Weighted Product* dalam Pemilihan Penerima Beasiswa Bagi Peserta Didik”. Pihak sekolah kesulitan dalam memberikan beasiswa kepada peserta didiknya, dimana penyebab permasalahan tersebut adalah banyaknya pilihan yang dapat diambil tetapi tidak adanya cara yang tepat dalam pemilihan keputusan tersebut. Dalam pemilihan penerima beasiswa memerlukan beberapa kriteria yaitu nilai rata-rata, tingkah laku, ekstrakurikuler, tanggungan orang tua itu merupakan *benefit* sedangkan pendapatan orang tua merupakan *cost*. Metode penelitian yang didalamnya berisikan kerangka pemikiran. Kerangkaan pemikiran gambaran dari langkah-langkah yang dilakukan pada saat melakukan penelitian agar penelitian berjalan secara sistematis[9].

Penelitian yang akan dilakukan memiliki perbedaan dengan penelitian sebelum-sebelumnya. Pada kesempatan kali ini peneliti sebagai penulis atau pengembang sistem akan membuat sistem pendukung keputusan rekrutmen pengurus HMTI PNC menggunakan metode *weighted product*. Sistem pendukung keputusan ini dibuat dengan sistem berbasis *website*, agar bisa diakses *online*. Dalam rekrutmen pengurus HMTI PNC diperlukan kriteria penilaian yang akan digunakan pada sistem, kriteria tersebut adalah pengalaman organisasi, psikologi, memiliki surat peringatan (SP) dan IPK. Sistem ini hanya bisa diakses oleh pengurus HMTI PNC atau panitia pemilihan dan calon pengurus HMTI PNC. Selain itu, sistem dapat menghasilkan laporan hasil perhitungan dari penilaian peserta dengan kriteria yang sudah ditentukan. Tahap pengembang sistem ini menggunakan model *waterfall* Sommerville yang dibangun menggunakan bahasa pemrograman PHP dan *MySQL*.

2.2 Landasan Teori

2.2.1 Sistem Pendukung Keputusan (*Decision Support System*)

Menurut Djamain dan Christin (2015), “Sistem pendukung keputusan (SPK) mengacu pada suatu sistem yang memanfaatkan dukungan komputer dalam proses pengambilan keputusan. Sistem pendukung keputusan (SPK) dirancang untuk mendukung seluruh tahap pengambilan keputusan mulai dari mengidentifikasi masalah, memilih data yang relevan, dan menentukan pendekatan yang digunakan dalam

proses pengambilan keputusan, sampai mengevaluasi pemilihan alternatif.”

Berdasarkan pengertian di atas diperoleh informasi bahwa sistem pendukung keputusan (SPK) bukan merupakan alat untuk pengambilan keputusan, melainkan merupakan sistem yang membantu pengambil keputusan dengan informasi dari data yang telah diolah dengan relevan dan diperlukan untuk membuat keputusan tentang suatu masalah dengan lebih cepat dan akurat. Sehingga sistem ini tidak dimaksudkan untuk menggantikan pengambilan keputusan dalam proses pembuatan keputusan.

2.2.1.1 Karakteristik Sistem Pendukung Keputusan

Karakteristik sistem pendukung keputusan:

1. Sistem pendukung keputusan dirancang untuk membantu pengambil keputusan dalam memecahkan masalah yang sifatnya semi terstruktur ataupun tidak terstruktur dengan menambahkan kebijaksanaan manusia dan informasi komputerisasi.
2. Dalam proses pengolahannya, sistem pendukung keputusan mengkombinasikan penggunaan model-model analisis dengan teknik pemasukan data konvensional serta fungsi-fungsi pencari/integrasi informasi.
3. Sistem pendukung keputusan, dirancang sedemikian rupa sehingga dapat digunakan/dioperasikan dengan mudah.
4. Sistem pendukung keputusan dirancang dengan menekankan pada aspek fleksibilitas serta kemampuan adaptasi yang tinggi[10].

2.2.2 Pengertian Metode *Weighted Product* (WP)

Metode *Weighted Product* (WP) adalah salah satu metode yang berfokus pada menyelesaikan masalah MADM (Multi Atribut Decision Making). Metode *Weighted Product* (WP) meneliti beberapa pilihan lain terhadap sekelompok perlengkapan atau tipe, dimana setiap perlengkapannya saling tidak bergantung satu sama lain.

Metode *Weighted Product* merupakan metode pengambilan keputusan dengan cara perkalian untuk menghubungkan ranting atribut, dimana ranting setiap atribut harus dipangkatkan dulu dengan bobot atribut yang bersangkutan dan menghasilkan nilai terbesar yang akan terpilih sebagai alternatif yang terbaik. Proses ini sama halnya dengan normalisasi. Perhitungan akan sesuai dengan metode ini apabila

alternatif yang terpilih memenuhi kriteria yang telah ditentukan. Metode *weighted product* (WP) ini lebih efisien karena waktu yang dibutuhkan dalam perhitungan lebih singkat[11][12].

2.2.2.1 Penerapan Metode *Weighted Product* (WP)

Menurut Nafi, Nihru, dkk (2017) setelah data dianalisis, kemudian data diproses dengan menggunakan metode *Weighted Product* (WP) dengan langkah sebagai berikut:

1. Menentukan kriteria - kriteria yang akan dijadikan acuan dalam pengambilan keputusan.
2. Menentukan rating kecocokan setiap alternatif pada setiap kriteria.
3. Menentukan bobot preferensi setiap kriteria.
4. Mengalikan seluruh atribut bagi sebuah alternatif dengan bobot sebagai pangkat positif dan bobot berpangkat negatif.
5. Hasil perkalian tersebut dijumlahkan untuk menghasilkan nilai V untuk setiap alternatif.
6. Mencari nilai alternatif dengan melakukan langkah yang sama seperti pada langkah satu, hanya saja menggunakan nilai tertinggi dan nilai terendah.
7. Membagi nilai V bagi setiap alternatif dengan nilai standar.
8. Mencari nilai alternatif ideal.

Penerapan metode *weighted product* (WP) digunakan dalam proses perhitungan nilai dari kriteria secara keseluruhan yang hasilnya dapat menentukan karyawan baru yang akan diterima.

2.2.2.2 Perhitungan Metode *Weighted Product* (WP)

Metode *weighted product* (WP) dalam perhitungannya lebih singkat yaitu terdiri dari 3 langkah yaitu[12]:

1. Perbaikan bobot kriteria, dengan persamaan berikut:

$$W_j = \frac{W_j}{\sum W_j}$$

2. Menghitung vektor S. Langkah ini sama seperti proses normalisasi, dengan persamaan berikut:

$$S_i = \prod_{j=1}^n Y_{ij}^{w_j} \quad \text{dengan } i=1,2,3,\dots$$

3. Menghitung vektor V, atau preferensi relatif dari setiap alternatif, untuk perankingan dengan persamaan berikut:

$$V_1 = \frac{S_1}{S_1 + S_2 + S_3 + \dots + S_i}$$

Keterangan:

S = Preferensi alternatif, dianalogikan sebagai vektor S.

V = Prefrensi alternatif dianalogikan sebagai vektor V.

W = Bobot kriteria.

i = Alternatif.

j = Kriteria.

n = Banyaknya kriteria.

. = Banyaknya kriteria yang telah di nilai pada vektor S.

Sedangkan untuk kriterianya terbagi dalam dua kategori yaitu untuk bernilai positif termasuk dalam kriteria keuntungan (*benefit*) dan yang bernilai negatif termasuk dalam kriteria biaya (*cost*).

2.2.3 Rekrutmen

Rekrutmen menurut Mathis dan Jakson (2001) adalah proses yang menghasilkan sejumlah pelamar yang berkualifikasi untuk pekerjaan di suatu perusahaan atau organisasi. Hal ini sejalan dengan pendapat Irawan dkk. (1997) bahwa rekrutmen adalah suatu proses untuk mendapatkan tenaga yang berkualitas guna bekerja pada perusahaan atau instansi. Sedangkan menurut ahli lain menyebutkan bahwa rekrutmen adalah proses mencari, menemukan, mengajak, dan menetapkan sejumlah orang, baik dari dalam maupun dari luar perusahaan sebagai calon tenaga kerja dengan karakteristik tertentu seperti yang telah ditetapkan dalam perencanaan SDM (Samsudin, 2006). Ivancevich (2001) menambahkan bahwa rekrutmen berkaitan dengan aktivitas yang mempengaruhi jumlah dan jenis pelamar, apakah pelamar tersebut kemudian menerima pekerjaan yang ditawarkan[13].

2.2.4 MySQL (*My Structure Query Language*)

MySQL (*My Structure Query Language*) adalah salah satu *DataBase Management System* (DBMS) dari sekian banyak DBMS seperti Oracle, MySQL, Postagre SQL, dan lainnya. MySQL berfungsi untuk mengolah database menggunakan bahasa SQL. MySQL bersifat open source sehingga kita bisa menggunakan secara gratis. Pemrograman PHP juga sangat mendukung/support dengan database MySQL [14].

2.2.5 HTML (*Hyper Text Markup Language*)

HTML (*Hyper Text Markup Language*) adalah sekumpulan simbol-simbol atau tag-tag yang dituliskan dalam sebuah file yang digunakan untuk menampilkan halaman pada *web browser*. Tag-tag HTML selalu diawali dengan `<x>` dan diakhiri dengan `</x>` dimana x tag HTML itu seperti b, i, u, dll [14].

2.2.6 Web Server

Web server adalah aplikasi yang berfungsi untuk melayani permintaan pemanggilan alamat dari pengguna melalui *web browser*, dimana *web server* mengirimkan kembali informasi yang diminta tersebut melalui HTTP (*HiperText Transfer Protocol*) untuk ditampilkan ke layar monitor komputer kita[14].

2.2.7 Codeigniter

Codeigniter merupakan sebuah konstruksi yang berguna untuk mengembangkan perangkat lunak berbasis *website*. Salah satu kelebihan *Codeigniter* adalah memudahkan pengguna dalam pemrograman bahasa PHP yang dinamis karena *Codeigniter* merupakan *framework* yang *open source*. Selain itu, *codeigniter* juga memudahkan pengembang untuk membuat perangkat lunak lebih cepat dari pada melakukan penulisan dari awal. *Codeigniter* memiliki banyak *library* yang dapat memudahkan dan membantu dalam proses pengembangan atau pembuatan perangkat lunak[15].

2.2.8 PHP (*Hypertext Preprocessor*)

PHP (*Hypertext Preprocessor*) yaitu bahasa pemrograman web server-side yang bersifat *open source*. PHP merupakan script yang terintegrasi dengan HTML dan berada pada server (*server side HTML embedded scripting*). PHP adalah script yang digunakan untuk membuat halaman *website* yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh *client*[14].

2.2.9 Website

Sebuah *website* biasanya dibangun atas banyak halaman web yang saling berhubungan. Jadi dapat dikatakan bahwa, pengertian *website* adalah kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak (animasi), suara, atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis

yang membentuk satu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaringan- jaringan halaman[16].

2.2.10 Rekayasa Perangkat Lunak

Perangkat lunak adalah seluruh Perintah (program komputer) yang bila dieksekusi memberikan fungsi dan unjuk kerja seperti yang diharapkan. Struktur data yang memungkinkan program memanipulasi informasi secara proporsional. Dokumen yang menggambarkan operasi dan kegunaan program.

A. Metode Pengembang Sistem

Dalam merancang dan membangun Sistem Pendukung Keputusan Rekrutmen Pengurus HMTI PNC Berbasis *Website* ini menggunakan metode *waterfall* Sommerville, karena metode ini memiliki tahapan yang sistematis dan sekuensial. Untuk tahapan pengembangnya sebagai berikut[4]:

1. *Requirement Definition*
Requirement Definition atau kebutuhan perlengkapan adalah tahapan penetapan fitur, kendala dan tujuan sistem melalui konsultasi dengan pengguna sistem. Semua hal tersebut akan ditetapkan secara rinci dan berfungsi sebagai spesifikasi sistem.
2. *System and Software Design*
Pada Tahap *System and Software Design* ini akan dibentuk suatu arsitektur sistem berdasarkan persyaratan yang telah ditetapkan. Selain itu juga, dilakukan identifikasi dan penggambaran terhadap abstraksi dasar sistem perangkat lunak beserta hubungan-hubungannya.
3. *Implementation and Unit Testing*
Dalam tahapan *Implementation and Unit Testing* ini, hasil dari desain perangkat lunak akan direalisasikan sebagai satu set program atau unit program. Setiap unit akan diuji apakah sudah memenuhi spesifikasinya.
4. *Integration and System Testing*
Dalam tahap *Integration and System Testing* ini, setiap unit program akan diintegrasikan satu sama lain dan diuji sebagai satu sistem yang utuh untuk memastikan sistem sudah memenuhi persyaratan yang ada. Setelah itu sistem akan dikirim ke pengguna sistem.

5. *Operation and Maintenance*

Dalam tahap *Operation and Maintenance* ini, sistem diinstal dan mulai digunakan. Selain itu juga memperbaiki error yang tidak ditemukan pada tahap pembuatan. Dalam tahap ini juga dilakukan pengembangan sistem seperti penambahan fitur dan fungsi baru.

B. Metode Pengujian Sistem

Pengujian *Black-box Testing* adalah pengujian yang dilakukan dan hanya fokus pada fungsionalitas dari spesifikasi *software*. Pendefinisian kondisi input dan pengetesan nantinya akan dilakukan oleh tester[17].

Ujicoba *blackbox* berusaha untuk menemukan kesalahan dalam beberapa kategori, diantaranya:

- 1) Fungsi-fungsi yang salah atau hilang
- 2) Kesalahan interface
- 3) Kesalahan dalam struktur data atau akses database eksternal
- 4) Kesalahan performa
- 5) kesalahan inisialisasi dan terminasi

2.2.11 Database Management System

Database Management System (DBMS) merupakan *software* yang digunakan untuk pengolahan, pembuatan, pemeliharaan, dan penggunaan data dengan skala yang besar. Selain itu DBMS juga bisa digunakan untuk melakukan manipulasi data. Karena hal-hal tersebut DBMS telah menjadi pemeran utama sebagai pendukung sebuah perusahaan. Salah satu contoh DBMS yang sering digunakan adalah Mysql dan MariaDB.

Dalam segala aspek penggunaan DBMS sangatlah penting. Contohnya pada media social seperti facebook yang menggunakan DBMS guna menyimpan sebuah data-data user facebook ke dalam DBMS.

2.2.12 Basis Data (Database)

Database atau Basis Data terdiri dari dua kata, yaitu Basis dan Data. Basis kurang lebih dapat diartikan sebagai markas atau gudang, tempat bersarang/berkumpul. Sedangkan data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa,

pembeli, pelanggan), barang, hewan, peristiwa, konsep, keadaan dan sebagainya yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya[10].

Jadi, dapat disimpulkan bahwa Basis Data merupakan suatu susunan data yang terstruktur dan saling berhubungan, disimpan dalam media penyimpanan elektronik/pengingat (*harddisk*) agar data dapat diakses secara mudah dan cepat. Basis Data untuk mengatasi beberapa masalah yang sering dihadapi ketika proses pengolahan data seperti:

1. Redudansi dan Inkonsistensi
Data penyimpanan yang sama pada beberapa tempat atau media penyimpanan yang akhirnya mengakibatkan terjadinya pemborosan media penyimpanan. Penyimpanan data yang sama dan berulang-ulang di beberapa file dapat mengakibatkan inkonsistensi.
2. Keamanan Data
Dengan *database management system*, sistem keamanan data bisa dicapai.
3. Kesulitan Mengakses Data
Database dapat mengakses kesulitan dalam mengakses data karena mampu mengambil data secara langsung dengan program aplikasi yang mudah digunakan.
4. Isolasi Data untuk Standarisasi
Jika data tersebar dalam bentuk format yang tidak sama, maka ini menyulitkan dalam menulis program aplikasi untuk mengambil dan menyimpan data. Maka suatu *database* haruslah dibuat suatu format, sehingga mudah dibuat program aplikasi.

Database dikelola dengan cara menuliskan sebuah kode perintah berupa *query SQL* yang harus dimasukkan sesuai kebutuhan dan hal tersebut terbagi menjadi dua yaitu: DDL (*Data Definition Language*) dan DML (*Data Manipulation Language*). Berikut di bawah ini merupakan pengertian DDL dan DML:

1. DDL
DDL digunakan untuk pengoprasian skema struktur pada *database*, perintah utama yang dapat digunakan pada DDL diantaranya *create*, *rename*, *alter*, dan *drop*.
 - a. *Create*
Perintah untuk membuat *database* dan *table*. Contoh perintah, "CREATE TABLE akun(id_akun INTEGER

PRIMARY KEY, jabatan VARCHAR(128) NOT NULL, email VARCHAR(128) NOT NULL, username VARCHAR(10) NOT NULL, password VARCHAR(8) NOT NULL, role_id INTEGER(10) NOT NULL);”

- b. *Rename*
Perintah yang digunakan untuk mengganti nama *table*. Contoh perintah, “RENAME TABLE akun TO akn;”
- c. *Alter*
Perintah yang digunakan untuk merubah atau mengedit. Contoh perintah, “Alter TABLE akn ADD tanggal Date; Alter TABLE akn Drop email;”
- d. *Drop*
Drop digunakan untuk menghapus *table* maupun *database*. Contoh perintah, “Drop Table akn;”

2. DML

DML merupakan perintah yang dipakai untuk melakukan pengolahan pada *database* seperti menambah, menghapus, membuat baru dan menampilkan. Perintah utama pada DML diantaranya adalah *select*, *insert*, *update* dan *delete*.

- a. *Select*
Perintah yang berfungsi untuk menampilkan data yang telah dimasukkan. Contoh perintah, “SELECT * FROM akn;”
- b. *Insert*
Perintah untuk memasukkan data ke dalam *database*. Contoh perintah, “INSERT INTO akn (id_akun, jabatan, email, username, password, role_id) values(P1, “MU”, “hanihidayati816@gmail.com”, “hani”, “hani123*”, “1”);”
- c. *Update*
Perintah yang dilakukan untuk pembaruan data. Contoh perintah, “UPDATE akn SET jabatan = “PSDM” WHERE id_akun = P1;”
- d. *Delete*
Perintah untuk menghapus data. Contoh perintah, “DELETE FROM akn WHERE username=“hani”;”

2.2.13 ERD (*Entity Relationship Diagram*)

ERD merupakan notasi grafis dalam pemodelan data konseptual yang mendeskripsikan hubungan antara penyimpanan. ERD digunakan untuk memodelkan struktur data dan hubungan antar data, karena hal ini

relatif kompleks. ERD menggunakan sejumlah notasi dan simbol untuk menggambarkan struktur dan hubungan antar data, pada dasarnya ada 3 macam simbol yang digunakan yaitu:

1. Entity : adalah suatu objek yang dapat diidentifikasi dalam lingkungan pemakai, sesuatu yang penting bagi pemakai dalam konteks sistem yang akan dibuat. Entiti digambarkan dalam bentuk persegi empat.
2. Atribut : Entiti mempunyai elemen yang disebut atribut, dan berfungsi mendeskripsikan karakter entiti. Misalnya atribut nama pekerja dari entiti pekerja. Setiap ERD bisa terdapat lebih dari satu atribut. Entiti digambarkan dalam bentuk ellips.
3. Hubungan : Relationship, sebagaimana halnya entiti maka dalam hubunganpun harus dibedakan antara hubungan atau bentuk hubungan antar entiti dengan isi dari hubungan itu sendiri. Misalnya dalam kasus hubungan antara entiti siswa dan entiti mata_kuliah adalah mengikuti, sedangkan isi hubungannya dapat berupa nilai_ujian. Relationship digambarkan dalam bentuk intan / diamonds[18].

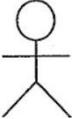
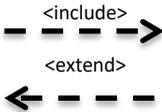
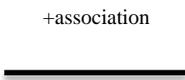
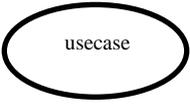
2.2.14 UML (*Unified Modelling Language*)

Unified Modeling Language (UML) adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem[10].

2.2.14.1 *Use Case Diagram*

Use case adalah pemodelan sistem informasi yang akan dibangun dengan mencantumkan kelakuan atau interaksi antara sistem dan *actor*. *Use case* digunakan untuk memberi informasi mengenai fungsi yang berada di dalam sistem dan siapa yang menggunakan fungsi-fungsi tersebut[10].

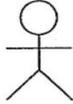
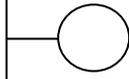
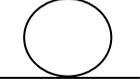
Tabel 2. 1 *Usecase Diagram*

No	Simbol	Fungsi
1		Segala sesuatu yang berinteraksi dengan sistem aplikasi komputer. Jadi <i>actor</i> ini bisa berupa orang, perangkat keras atau mungkin juga obyek lain dalam sistem yang sama.
2		Menspesifikasikan bahwa perilaku <i>use case</i> merupakan bagian dari <i>use case</i> lain.
3		Menggambarkan navigasi antar <i>class</i> , berupa banyak obyek lain yang berhubungan dengan satu obyek, dan apakah suatu <i>class</i> menjadi bagian dari <i>class</i> lainnya.
4		<i>System Boundary</i> yaitu batasan sebuah sistem.
5		<i>Use case</i> menjelaskan urutan kegiatan yang dilakukan aktor dan sistem untuk mencapai suatu tujuan tertentu. Walaupun menjelaskan kegiatan, <i>use case</i> hanya menjelaskan apa yang dilakukan oleh aktor dan sistem, bukan bagaimana aktor dan sistem melakukan kegiatan.

2.2.14.2 *Sequence Diagram*

Sequence diagram merupakan diagram yang menjelaskan behavior objek ke *use case*. Kelakuan objek nanti akan dijelaskan dengan mendeskripsikan jangka hidup objek dan pesan yang diterima dan dikirim[10].

Tabel 2. 2 *Sequence Diagram*

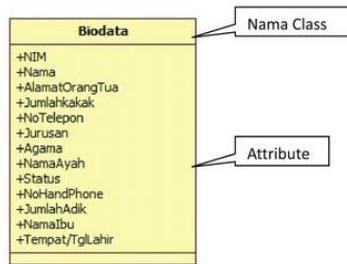
No	Simbol	Fungsi
1		<i>Objek entity</i> , antarmuka yang saling berinteraksi.
2		Digunakan untuk menggambarkan <i>user/Pengguna</i> .
3	Message() 	Spesifikasi dari komunikasi antar objek yang memuat informasi – informasi tentang aktifitas yang terjadi.
4		Digunakan untuk menggambarkan sebuah <i>form</i> .
5		Digunakan untuk menghubungkan <i>Boundary</i> dengan tabel.
6		Digunakan untuk menggambarkan hubungan kegiatan yang akan dilakukan.

2.2.14.3 Class Diagram

Dalam UML, ada dua jenis *class diagram* yaitu:

1. *Domain class diagram*

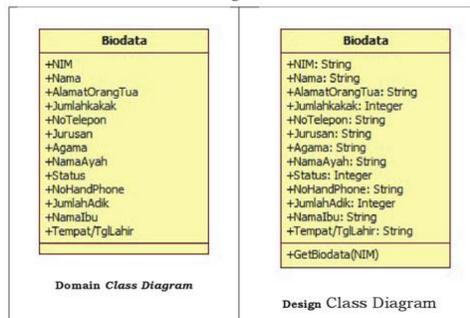
Domain class diagram adalah pada sesuatu dalam lingkungan kerja pengguna, bukan pada class perangkat lunak yang nantinya akan anda rancang.



Gambar 2. 1 Domain Class Diagram

2. Design class diagram

Tujuan utamanya adalah untuk mendokumentasikan dan menggambarkan kelas-kelas dalam pemrograman yang nantinya akan dibangun. *Desain class diagram* menggambarkan kelas berorientasi objek yang dibutuhkan dalam pemrograman, navigasi di antara kelas, *attribute names*, dan propertinya, serta *method names* dan propertinya. Gambar di bawah ini menunjukkan *domain class diagram* yang dikembangkan pada tahap analisis dan versi *design class diagram*. Versi *design class diagram* memiliki sebuah kompartemen baru di bagian bawah yang menentukan *method signatures*[19].



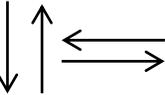
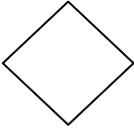
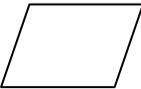
Gambar 2. 2 Domain Class Diagram menjadi Design Class Diagram

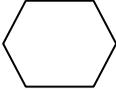
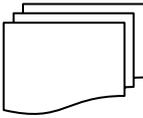
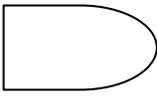
2.2.15 Flowchart

Flowchart atau bagan alir adalah penggambaran secara grafik dari langkah-langkah dan urutan prosedur dari suatu program. Tujuan *flowchart* merupakan untuk membantu menggambarkan suatu alur dalam menyelesaikan masalah dengan menggunakan simbol-simbol.

Analisis sistem menggunakan *flowchart* program untuk menggambarkan urutan tugas-tugas pekerjaan dalam suatu prosedur atau operasi. Berikut ini simbol-simbol di dalam *flowchart*[10].

Tabel 2. 3 *Flowchart*

No	Gambar Simbol	Nama	Keterangan
1		<i>Terminal point</i>	Menunjukkan mulai atau berakhirnya suatu proses.
2		<i>Flow direction</i>	Simbol yang digunakan untuk menghubungkan antar simbol, berfungsi juga untuk menunjukkan alur sebuah proses.
3		Proses	Menunjukkan sebuah kegiatan yang dilakukan oleh komputer atau sistem.
4		<i>Decision</i>	Simbol yang digunakan untuk memilih sebuah keputusan atau proses sesuai dengan kondisi yang ada.
5		<i>Input output</i>	Menunjukkan sebuah <i>input</i> atau <i>output</i> yang ada tidak bergantung pada jenis peralatannya.
6		<i>Predefined process</i>	Menunjukkan pelaksanaan suatu bagian prosedur, bagian prosedur yang terinformasi belum detail dan akan diperinci ditempat lain.
7		<i>Connector (On-page)</i>	Menghubungkan suatu simbol pada satu halaman yang letaknya berjauhan.

8		<i>Connector (Off-page)</i>	Menghubungkan simbol yang berada dalam halaman yang berbeda.
9		<i>Preparation</i>	Simbol ini menunjukkan persiapan penyimpanan kedalam <i>storage</i> .
10		<i>Manual input</i>	Menunjukkan proses <i>input</i> yang dilakukan secara <i>manual</i> menggunakan <i>online keyboard</i> .
11		<i>Manual operation</i>	Menunjukkan proses/kegiatan yang dilakukan tanpa menggunakan komputer.
12		<i>Document</i>	Menunjukkan <i>input</i> berupa dokumen dalam bentuk kertas atau <i>output</i> yang harus dicetak.
13		<i>Multiple document</i>	Sama seperti simbol <i>document</i> hanya saja dokumen yang digunakan lebih dari satu.
14		<i>Display</i>	Simbol yang menunjukkan adanya penggunaan peralatan <i>output</i> .
15		<i>Delay</i>	Menunjukkan bahwa adanya proses <i>delay</i> .

-{halaman sengaja dikosongkan}-