

BAB II LANDASAN TEORI

1.1 Tinjauan Pustaka

Dalam penelitian ini akan digunakan lima tinjauan pustaka yang nantinya dapat mendukung penelitian, berikut ini merupakan tinjauan pustaka yang diambil yaitu:

Tabel 2 1 Tinjauan Pustaka

No.1	(Asrul Abdullah, 2018)
Judul	Perancangan Sistem Pendukung Keputusan Dalam Pemilihan Dosen Pembimbing Skripsi Berdasarkan Minat Mahasiswa dengan Metode AHP (Analytical Hierarchy Process) di Universitas Muhammadiyah Pontianak
Jurnal	Edukasi dan Penelitian Informatika
Volume & Halaman	Vol. 4 & Hal. 184 – 191
Tahun	2018
Penulis	Asrul Abdullah dan Menur Wahyu Pangestika
Identifikasi Masalah	pemilihan dosen pembimbing skripsi diputuskan oleh pihak program studi lewat rapat para dosen untuk menentukan dosen mana yang sesuai dengan judul dan deskripsi dari mahasiswa yang mengambil mata kuliah skripsi tersebut.
Metode / Tools	<i>AHP (Analytical Hierarchy Process)</i>
Hasil penelitian	sistem memberikan ranking tiap alternatif. Ranking pertama menjadi prioritas pertama sistem menentukan keputusan dengan melihat kecocokan minat mahasiswa.
No. 2	(Bosker Sinaga, 2018)
Judul	Sistem Pendukung Keputusan Penentuan Dosen Pembimbing Skripsi Menggunakan Metode Profile Matching (Studi Kasus : Stmik Pelita Nusantara Medan)
Jurnal	Mantik Penusa
Volume & Halaman	Vol. 2 & Hal. 71 – 79
Tahun	2018
Penulis	Bosker Sinaga dan Yulia Utami
Identifikasi Masalah	Dalam penentuan dosen pembimbing harus sangat diperhatikan terlebih kesesuaian dengan bidangnya. Penentuan dosen pembimbing ditentukan oleh Ka.Prodi, pekerjaan harus sangat diperhatikan sehingga sangat memerlukan waktu dalam penyesuaian judul skripsi dan kepakaran dosen.
Metode / Tools	<i>Profile Matching</i>

Hasil penelitian	Hasil akhir penelitian ini adalah menghasilkan sistem pendukung keputusan penentuan dosen pembimbing skripsi dengan menerapkan metode profile matching
No. 3	(Nono Sudarsono, 2021)
Judul	Sistem Pendukung Keputusan Penentuan Dosen Pembimbing Dan Bimbingan Thesis Secara Online Pada Program Pascasarjana Universitas Galuh
Jurnal	SISTEM INFORMASI DAN TEKNOLOGI INFORMASI
Volume & Halaman	Vol. 10 & Hal. 15 – 29
Tahun	2021
Penulis	Nono Sudarsono, Harin Sanditha Rahmawati
Identifikasi Masalah	Dalam penentuan pembimbing thesis, dosen juga harus membaca setiap judul untuk benar-benar mencocokkan dengan keahlian dosen dan pengalaman dosen dalam membimbing. Rapat dapat memakan waktu sehari kerja untuk menentukan pembimbing untuk lebih dari 50 mahasiswa, ini sangat tidak efisien.
Metode / Tools	<i>Software Development Life Cycle</i>
Hasil penelitian	Hasil yang dicapai dari sistem pendukung keputusan ini dapat membantu proses pemilihan dosen pembimbing menjadi lebih cepat, menghasilkan nilai CR = 0,07 yang menunjukkan konsistensi nilai perbandingan kriteria yang diberikan dan mampu menghasilkan dosen pembimbing dalam melakukan bimbingan thesis secara online.
No. 4	(Marsani Asfi, 2020)
Judul	Implementasi Algoritma Naive Bayes Classifier sebagai Sistem Rekomendasi Pembimbing Skripsi
Jurnal	Nasional Informatika dan Teknologi Jaringan
Volume & Halaman	Vol. 5 & Hal. 44 – 50
Tahun	2020
Penulis	Marsani Asfi Dan Nopi Fitrianiingsih
Identifikasi Masalah	Salah satu permasalahan yang terjadi adalah pada tahap penentuan dosen pembimbing skripsi yaitu masih ditentukan berdasarkan subjektifitas dan kebijakan ketua program studi. Penentuan dosen pembimbing terkadang belum dilakukan penyesuaian antara bidang ilmu skripsi yang dipilih mahasiswa dengan kompetensi dosen yang bersangkutan.
Metode / Tools	<i>Algoritma Naive Bayes Classifier</i>
Hasil penelitian	Berdasarkan hasil pengujian algoritma Naive Bayes Classifier didapatkan perbandingan tingkat kesesuaian dosen pembimbing 1 sebesar 90% : 10% dan perbandingan tingkat kesesuaian dosen pembimbing 2 sebesar 30% : 70%.
No. 5	(Tantri Hari Mukti, 2018)

Judul	Sistem Pemilihan Dosen Pembimbing Skripsi Dengan Metode Topsis (Studi Kasus : Pendidikan Teknik Informatika)
Jurnal	Ilmiah Flash
Volume & Halaman	Vol. 4 & Hal. 53 – 63
Tahun	2018
Penulis	Tantri Hari Mukti, Syaad Patmantara, dan Aji Prasetya Wibawa
Identifikasi Masalah	Penentuan dosen pembimbing skripsi yang sekarang dilakukan oleh Tim KBK dan Korprodi mempunyai keterbatasan yaitu dengan sekian banyak jumlah mahasiswa beserta judul yang diajukannya, Tim KBK dan Korprodi akan kesulitan memilah dosen yang sesuai dengan judul masing-masing mahasiswa.
Metode / Tools	<i>Algoritma Topsis dan Metode Waterfall</i>
Hasil penelitian	Pemilihan metode TOPSIS (Technique For Others Reference by Similarity to Ideal Solution) dirasa lebih cocok dengan kasus ini karena terdapatnya pemecahan unsur kriteria dan pembobotan pada masing-masing kriteria.

1.2 Sistem Pendukung Keputusan (SPK)

“Sistem pendukung keputusan memberikan dukungan kepada seorang manajer dalam memecahkan masalah semi terstruktur dengan memberikan informasi atau saran mengenai keputusan tertentu, informasi tersebut dapat diberikan dalam bentuk laporan berkala, laporan khusus maupun model matematis” (Diana, 2018). Sistem ini mempresentasikan permasalahan manajemen yang dihadapi sehari-hari ke dalam bentuk kuantitatif, misalnya dalam bentuk model matematika.

1.3 Metode SAW

Metode SAW sering juga dikenal dengan istilah metode penjumlahan terbobot. Konsep dasar metode SAW adalah mencari penjumlahan terbobot dari rating kinerja pada setiap alternatif pada semua atribut. Metode SAW membutuhkan proses normalisasi matriks keputusan (X) ke suatu skala yang dapat diperbandingkan dengan semua rating alternatif yang ada.

$$r_{ij} = \frac{X_{ij}}{\text{Max}_i X_{ij}}$$

jika j adalah atribut keuntungan (benefit)

$$r_{ij} = \frac{\text{Min}_i X_{ij}}{X_{ij}}$$

jika j adalah atribut biaya (cost)

Dimana r_{ij} adalah rating kinerja ternormalisasi dari alternatif (A_i) pada atribut C_j ; $i=1,2,\dots,m$ dan $j=1,2,\dots,n$. Nilai preferensi untuk setiap alternatif (V_i) diberikan sebagai:

$$V_i = \sum_{j=1}^n W_j r_{ij}$$

Nilai yang lebih besar mengindikasikan bahwa alternatif lebih terpilih. Adapun langkah-langkahnya adalah:

1. Memberikan nilai setiap alternatif (A_i) pada setiap kriteria (C_j) yang sudah ditentukan, dimana nilai tersebut diperoleh berdasarkan nilai crisp; $i=1,2,\dots,m$ dan $j=1,2,\dots,n$.
2. Memberikan nilai bobot (W) yang juga didapatkan berdasarkan nilai crisp.
3. Melakukan normalisasi matriks dengan cara menghitung nilai rating kinerja ternormalisasi (r_{ij}) dari alternatif (A_i) pada atribut (C_j) berdasarkan persamaan yang disesuaikan dengan jenis atribut (atribut keuntungan / benefit = MAKSIMUM atau atribut biaya / cost = MINIMUM). Apabila berupa atribut keuntungan maka nilai crisp (X_{ij}) dari setiap kolom atribut dibagi dengan nilai crisp MAX ($\text{MAX } X_{ij}$) dan tiap kolom, sedangkan untuk atribut biaya, nilai crisp MIN ($\text{MIN } X_{ij}$) dari tiap kolom atribut dibagi dengan nilai crisp (X_{ij}) setiap kolom.

4. Melakukan proses perangkingan dengan cara mengalikan matriks ternormalisasi (R) dengan nilai bobot (W).
5. Melakukan nilai preferensi untuk setiap alternatif (V_i) dengan cara menjumlahkan hasil kali antara matriks ternormalisasi (R) dengan nilai bobot (W). Nilai V_i yang lebih besar mengindikasikan bahwa alternatif (A_i) lebih terpilih.

1.4 Pengertian Website

Website atau situs web Josi, (2016), dapat diartikan sebagai kumpulan halaman yang menampilkan informasi data teks, data gambar, diam, atau gerak, data animasi, suara, video dan gabungan dari semuanya baik yang bersifat statis maupun dinamis yang membentuk suatu rangkaian bangunan saling terkait dimana masing-masing dihubungkan dengan jaringan halaman (hyperlink) yang dapat diakses melalui perangkat lunak yang disebut browser (perambah) adalah aplikasi yang mampu menjalankan dokumen-dokumen *web* dengan cara diterjemahkan. Prosesnya dilakukan oleh komponen yang terdapat didalam aplikasi browser yang biasa disebut *web engine* semua dokumen *web* ditampilkan dengan cara diterjemahkan. Beberapa contoh *web browser* yang populer saat ini adalah *Google Chrome*, *Mozilla Firefox*, dan *Opera*. Terdapat 2 jenis website, yaitu :

1.4.1 Website Statis

Merupakan *website* yang bersifat tidak mudah diubah isinya oleh pengguna. Untuk melakukan perubahan konten, pengguna harus merubahnya dengan cara *coding* pada halaman *website* atau merubah melalui *database*.

1.4.2 Website Dinamis

Merupakan *Website* yang mudah beradaptasi dan otomatis menyesuaikan

perubahan konten secara langsung tanpa harus merubah struktur kode *website*.

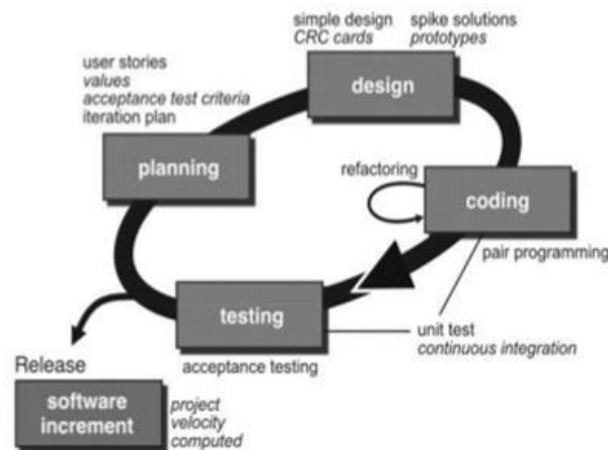
1.5 Metode Pengembangan Sistem

Pada Penelitian ini, penulis menggunakan metode *eXtreme Programming* (XP) sebagai metode pengembangan sistemnya, dengan menerapkan metode pengembangan sistem ini sangat cocok digunakan dalam penelitian, karena untuk membuat sistem memerlukan waktu yang singkat, dan juga metode ini sangat terstruktur.

2.5.1 eXtreme Programming (XP)

eXtreme Programming (XP) adalah metode pengembangan perangkat lunak yang banyak digunakan oleh developer. XP diperkenalkan oleh Kent Beck ketika ia ditunjuk untuk menangani sebuah proyek penggajian C3 (Chrysler Comprehensive Compensation). Proyek ini dimulai sekitar tahun 1996, proyek tersebut terancam gagal karena rumitnya sistem yang dibuat dan mendapatkan kegagalan pada saat memasuki proses uji sistem. XP terasa seperti sekumpulan konsep yang sudah ketinggalan zaman dan tidak banyak berpengaruh pada pengembangan aplikasi. Namun, XP memiliki keuntungan karena sesuai untuk proyek dengan beberapa kebutuhan pelanggan yang bertentangan XP juga idea untuk proyek pengembangan yang perlu beradaptasi dengan cepat terhadap perubahan selama pengembangan aplikasi. (Suryantara, 2017).

2.5.2 Kerangka Kerja *eXtreme Programming (XP)*



Gambar 2. 1 Tahapan Metode eXtreme Programming

Sumber: (Suryantara, 2017)

Pendekatan XP sering digunakan oleh programmer untuk membuat aplikasi dengan cepat. Tahapan yang meliputi perencanaan, perancangan, pengkodean, dan pengujian pengembangan perangkat lunak termasuk dalam metodologi XP.

2.5.3 Tahapan Perencanaan (*Planning*).

Tahapan awal ini merupakan langkah dalam pembangunan sistem. Pada tahap ini, menjelaskan spesifikasi kebutuhan (*System Requirement*) dari sistem yang akan dibuat. Tahap ini mendefinisikan keluaran (*output*), yang ada pada aplikasi, fungsi dari aplikasi yang dibuat, waktu dan biaya pengembangan aplikasi, serta alur pengembangan aplikasi. Berdasarkan hasil dari *user stories*, adapun story yang dilakukan dengan cara sebagai berikut.

1. Menceritakan permasalahan sistem yang digunakan seperti apa yang akan dibangun.
2. Berdasarkan hasil cerita pengguna menentukan poin pada bagian *value* untuk memutuskan apa saja yang akan dibangun
3. Dari hasil kesepakatan tersebut peneliti menentukan *acceptance test criteria* yaitu menentukan kriteria-kriteria nantinya sebagai acuan terhadap Sistem yang akan di uji.

4. Sehingga hasil peneliti menyimpulkan berapa kali akan dilakukan *realies* dan perbaikan pada tahapan *iteration plan* berapa kali akan dilakukan uji terhadap sistem yang dibangun.

2.5.4 Tahapan Perancangan (*Design*).

Tahapan kedua adalah perancangan dimana pada tahapan ini dilakukan kegiatan pemodelan yang dimulai dari pemodelan sistem, pemodelan arsitektur sampai dengan pemodelan basis data. *eXtreme Programming* pada proses pembuatan desain memberikan informasi gambaran sistem yang akan dibangun, beberapa desain yang akan dilakukan oleh peneliti:

1. *Simple Design CRC Card* untuk mengenali dan mengatur *object oriented class* yang sesuai dengan pengembangan. Jika pada saat tampilan terdapat ketidaksesuaian perbaikan maka akan dilakukan.
2. *Spike solution* yang dilakukan kepada pengguna mendapatkan kesesuaian antara keinginan pengguna dan pengembangan yang dilakukan.

2.5.5 Tahapan Pengkodean (*Coding*)

Tahapan ketiga dalam *eXtreme Programming* yaitu penerapan pengkodean yang sudah dibuat kedalam bentuk *user interface* dengan menggunakan bahasa pemrograman. Pada proses pengkodean peneliti menyesuaikan terhadap story pengguna sehingga, sistem yang dibangun dalam proses pengkodean yang dilakukan yaitu.

1. *Pair Programming* merupakan tahapan sistem dibangun dengan bahasa pemrograman dan media penyimpanan yang telah disepakati.
2. *Refactory* merupakan ketidaksesuaian kode program sehingga dilakukan perbaikan mendapatkan hasil yang sesuai.

2.5.6 Tahapan Pengujian (*Testing*).

Setelah tahapan ketiga (pengkodean) selesai, selanjutnya adalah tahapan terakhir yaitu

tahapan pengujian, tahapan pengujian ini dilakukan untuk untuk mengetahui kesalahan (error) apa saja yang muncul saat aplikasi sedang berjalan serta mengetahui apakah sistem yang dibangun sudah sesuai dengan kebutuhan pengguna. Metode pengujian yang digunakan pada tahapan ini adalah metode *is0 25010 testing*, dimana pengujian yang dilakukan terhadap form beberapa masukkan apakah sudah berjalan sesuai dengan fungsinya masing-masing. Tahapan pengujian dilakukan oleh pengguna sebagai pelanggan dan admin dengan melakukan uji sesuai dengan *acceptance test* yang telah ditentukan dan disetujui. Unit test yang telah dibuat fokus pada keseluruhan fitur dan fungsional sistem. Sehingga sistem dapat disimpulkan telah sesuai dan dapat di *realies*. Metode pengujian sistem merupakan metode yang digunakan untuk melakukan testing pada sistem yang dibangun sehingga diperoleh hasil berupa sistem yang sesuai fungsinya. Adapun dibawah ini adalah kelebihan dan kekurangan metode *eXtreme Programming (XP)*.

Kelebihan metode *eXtreme Programming (XP)*.

1. Dalam hal ini *eXtreme Programming (XP)* menjalin komunikasi dan interaksi secara baik dengan klien pada pengembangan aplikasi.
2. Setiap umpan balik langsung ditanggapi dengan melakukan test yang akurat mengenai aplikasi yang dibuat.
3. Dapat memberikan ide baru atau mengubah pemikiran klien terhadap sistem yang dibuat, sehingga memiliki keberanian untuk mencobanya.
4. Developer tidak bekerjasecara berlebihan, dikarenakan XP memfokuskan pda aplikasi/program yang dibuat.
5. Membuat keputusan-keputusan yang teknikal.

Kekurangan metode *eXtreme Programming (XP)*.

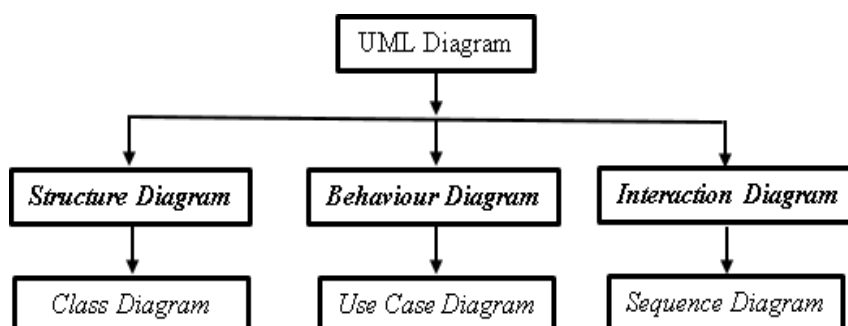
1. Perubahan selalu diterima sehingga developer harus siap dengan perubahan-perubahan dari klien.
2. Di awal-awal pengembangan aplikasi tidak dapat membuat kode (program) yang detail karna prinsip simplicity.
3. XP belum terbukti dapat bekerja pada sistem yang mempunyai masalah skalabilitas.

1.6 Unified Modelling Language (UML)

Metode pengembang sistem merupakan metode yang digunakan sebagai alur proses dalam pengembangan, sehingga penelitian dapat di kembangkan sesuai tahapan dari metode pengembang sistem.

Alat pengembang sistem merupakan konsep desain yang digunakan untuk menggambarkan sistem dengan menggunakan diagram. Penyesuaian alat yang digunakan harus sesuai dengan metode pengembangan yang dilakukan salah satunya adalah penerapan *Unified Modelling Language*. Rosa dan Salahuddin (2019), *Unified Modelling Language* adalah bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. Berikut ini merupakan penjelasan tentang masing-masing diagram yang ada pada *Unified Modelling Language*.

Gambar 2. 2 Pemodelan UML




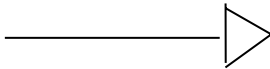


1.6.1 Use Case Diagram

Menurut (Rosa dan Salahuddin 2019) *use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. *Use case* digunakan untuk

mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Berikut simbol-simbol yang akan digunakan dalam menggambarkan *Use Case Diagram* dapat dilihat pada Tabel berikut ini:

Tabel 2 2 Simbol use case diagram

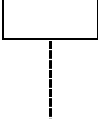


No	Simbol	Deskripsi
1.		<i>Use case</i> : Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i> .
2.		Aktor: seseorang/ sesuatu yang berinteraksi dengan yang akan dibuat. diluar sistem informasi. Biasanya dinyatakan menggunakan kata benda
3.		Asosiasi (<i>association</i>): merupakan 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.		Generalisasi (<i>generalization</i>): merupakan hubungan (umum – khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum
5.	<< <i>Include</i> >>	Include berarti <i>use case</i> yang ditambahkan akan dipanggil saat <i>use case</i> tambahan dijalankan.

6.	<<Extend>>	Ekstensi (extend) merupakan use case tambahan ke sebuah use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan itu.
----	------------	--

1.6.2 Sequence Diagram

Sequence Diagram rangkaian menggambarkan bagaimana objek berinteraksi dengan satu sama lain melalui pesan pada eksekusi sebuah *use-case* atau operasi. Diagram ini mengilustrasikan bagaimana pesan terkirim dan diterima di antara objek dan dalam sekuensi.

Tabel 2 3 Simbol Squence Diagram



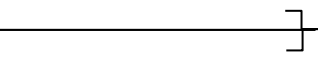
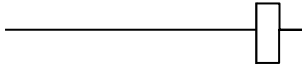
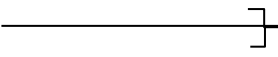

No.	Simbol	Deskripsi
1.	<p data-bbox="308 1160 496 1193"><i>Object lifeline</i></p> 	Menggambarkan panjang kehidupan suatu objek selama scenario sedang di buat contohnya
2.	<p data-bbox="308 1406 443 1440"><i>Activation</i></p> 	Dimana proses sedang dilakukan oleh <i>object</i> atau <i>class</i> untuk memenuhi pesan atau perintah
3.	<p data-bbox="308 1608 424 1641"><i>Message</i></p> 	Sebuah anak panah yang mengindikasikan Pesan diantara objek. Dan objek dapat mengirimkan pesan ke dirinya sendiri

1.6.3 Class Diagram

Rosa dan Salahuddin (2019) *Class diagram* mengembangkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Berikut simbol - simbol

yang akan digunakan dalam menggambarkan *Class Diagram* dapat dilihat pada Tabel berikut ini:

Tabel 2 4 Simbol Class Diagram

No.	Simbol	Deskripsi
1.	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> <p>Nama_kelas</p> <hr/> <p>+Attribute</p> <hr/> <p>+Operasi</p> </div>	Kelas pada struktur sistem.
2.	<p>Antar Muka/Interface</p>  <p>Nama_Interface</p>	Sama dengan konsep interface dalam pemrograman berorientasi objek.
3.	<p>Asosiasi / Association</p> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan symbol
4.	<p>Asosiasi Berarah / Directed Association</p> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan symbol.
5.	<p>Generalisasi</p> 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
6.	<p>Ketergantungan / dependency</p> 	Relasi antar kelas dengan makna ketergantungan antar kelas.
7.	<p>Agregasi / aggregation</p> 	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>)

1.7 Pengertian Pemrograman Berorientasi Objek

Pemrograman yang berorientasikan kepada objek, dimana semua data dan fungsi dibungkus dalam class - class atau object - object. Setiap object dapat menerima pesan, memproses data, mengirim, menyimpan dan memanipulasi data. Beberapa object berinteraksi dengan saling memberikan informasi satu terhadap yang lainnya. Penggunaan pemrograman berorientasi objek sangat banyak sekali, contoh : java, php, perl, c#, cobol, dan lainnya.

1.7.1 Basis Data

Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi tersedia saat dibutuhkan. Pada intinya basis data adalah media untuk penyimpanan data agar dapat diakses dengan mudah dan cepat". (A. S., 2018).

1.7.2 MySQL

Menurut (Anggraini, 2020) MySQL adalah nama database server. Database server adalah suatu pengorganisasian data dengan tujuan memudahkan penyimpanan dan pengaksesan data. MySQL tergolong sebagai database relasional. pada model ini, data dinyatakan dalam bentuk dua dimensi yaitu secara khusus dinamakan tabel, tabel tersusun atas baris dan kolom

1.7.3 Xampp

Menurut (Mawaddah U, 2018) XAMPP adalah software yang di dalamnya terdapat *server MySQL* dan didukung oleh *PHP* sebagai bahasa pemrograman untuk membuat website dinamis serta terdapat web server apache yang dapat dijalankan di beberapa *platform* seperti *OS X, Windows, Linux, Mac, dan Solaris*.

1.7.4 PHP

Menurut (Heryanto dan Lestari 2019) *Hypertext Preprocessor (PHP)* adalah satu bahasa pemrograman open source yang sangat cocok atau dikhususkan untuk pengembangan web dan dapat ditanamkan pada sebuah skripsi HTML. Bahasa *PHP* dapat dikatakan menggambarkan beberapa bahasa pemrograman seperti *C, Java, dan Perl* serta mudah untuk dipelajari.

1.7.5 Astah Community

Astah Community merupakan perangkat lunak yang penulis gunakan untuk membuat model UML. Astah dibuat oleh perusahaan Jepang bernama Change Vision. Astah terasa “ringan” bahkan untuk pemula sekalipun, hal ini di karenakan antar - mukanya yang user - friendly sehingga memungkinkan pengguna membuat diagram UML dengan mudah dan cepat.

Astah Community merupakan *tool* gratis dengan fitur-fitur dasar, dilengkapi dengan fitur mencetak diagram, mengimpor/mengekspor ke *program java*. *Astah Community* yang dulunya bernama *Astah Jude* telah digunakan sedikitnya 120.000 orang diseluruh dunia pada tahun 2006 (wikipedia.org). Perangkat lunak ini dapat diunduh melalui <http://astah.net/download>.

1.7.6 Microsoft Visual Studio



Gambar 2. 3 Logo VS Code

Microsoft Visual Studio Code adalah sebuah *tools editor code* yang didefinisikan ulang dan dioptimalkan untuk membangun dan men-debug aplikasi web dan cloud modern. *Microsoft Visual Studio Code* adalah *one-stop shop* yang memungkinkan kita fokus pada proses pengembangan (Sholihin, 2010). Berikut adalah beberapa fitur yang dimiliki *Microsoft Visual Studio Code*:

a. *Cross Platform*

Tersedia pada system operasi macOS, Linux dan Windows, yang artinya dapat bekerja pada system operasi manapun tanpa khawatir belajar *coding tools* yang sama untuk system operasi yang berbeda.

b. *Lightweight*

Fitur yang dapat mengontrol seluruh Bahasa, tema, *debugger*, *command* dan lain-lain

sesuai keinginan *editor*.

c. *Powerful Editor*

Memfungsikan fitur untuk *source code editing* yang sangat produktif.

d. *Code Debugging*

Fitur yang dapat membantu *editor* dalam melakukan *debugging* pada *code* dengan cara mengawasi *code*, *callstack*, *variable* dan *expression*.

e. *Source Control*

Microsoft Visual Studio Code memiliki *integrated source control* termasuk *Git support* dan penyedia *source code* lainnya.

f. *Integrated Terminal*

Fitur *command-line task* yang dapat membantu *editor*.