

BAB II

DASAR TEORI

2.1 Tinjauan Pustaka

Penelitian sebelumnya dilakukan di SMK Cengkareng 1. Penelitian ini dilakukan karena proses penerimaan siswa baru yang dilakukan oleh SMK Cengkareng 1 masih dilakukan secara konvensional, dimana calon siswa datang ke sekolah untuk melakukan pendaftaran. Sistem penerimaan siswa baru yang ada masih belum efisien karena formulir yang sudah diisi oleh calon siswa harus ditulis kembali ke dalam *Microsoft Excel*, serta dokumen yang banyak menyulitkan panitia dalam melakukan pencarian data. Berdasarkan permasalahan tersebut, maka dibuat sebuah sistem yang dapat melakukan proses penerimaan siswa baru yang tidak dibatasi oleh tempat menggunakan teknologi *website* dan menerapkan metode *waterfall* yang memungkinkan pembuatan sistem dilakukan secara teratur dan sistematis atau berurutan sesuai dengan siklus pengembangan yang ada [4].

Penelitian lainnya dilakukan di SMK Ma'arif Cicalengka. Penelitian ini dilakukan karena SMK Ma'arif Cicalengka belum memanfaatkan perkembangan teknologi secara maksimal menyangkut segala keperluan sekolah serta masih dilakukan secara konvensional, salah satunya pendaftaran siswa baru dimana para calon siswa dan walinya diharuskan datang langsung ke sekolah untuk mengambil formulir dan mengumpulkan berkas pendaftaran. Hal itu mengakibatkan adanya kendala karena terjadi penumpukkan permintaan siswa baru yang akan mendaftar sehingga pihak sekolah kewalahan. Oleh karena itu, dibuatlah aplikasi penerimaan siswa baru berbasis *website* dengan menggunakan model pengembangan *waterfall* sebagai pengembangan sistem dan dalam membangun *website* menggunakan *framework codeigniter* [5].

Penelitian sebelumnya juga pernah dilakukan di SDI Onekore 5. Penelitian ini dilakukan karena situasi pandemi saat ini membuat teknologi informasi sangat dibutuhkan oleh masyarakat. PPDB yang biasanya dilakukan dengan cara orang tua dan calon siswa baru datang ke sekolah, di masa pandemi hal tersebut tentu menjadi sangat berbahaya karena menyebabkan kerumunan banyak orang di lingkungan sekolah. Proses penerimaan siswa baru di SDI Onekore 5 masih dilakukan dengan mengisi data siswa baru menggunakan lembaran formulir pendaftaran

yang disediakan oleh panitia serta penyimpanan data siswa baru belum terkomputerisasi sehingga menimbulkan berbagai permasalahan. Dalam mengatasi kesulitan yang terjadi, maka harus dibuat aplikasi proses penerimaan siswa baru yang lebih efisien untuk meningkatkan efektivitas dan proses pengolahan data siswa tidak terjadi kesalahan [6].

Penelitian lainnya pernah dilakukan di SMK Bandara. Penelitian ini dilakukan karena sistem Penerimaan Siswa Baru (PSB) di SMK Bandara selama ini masih melalui formulir yang disebar antarpada calon siswa sehingga informasi PSB tidak dapat diakses banyak orang dan membutuhkan waktu yang lama untuk menyebarkan brosur-brosur tersebut. Pengisian formulir menggunakan brosur PSB yang masih menggunakan tulisan tangan, menyebabkan panitia kesulitan untuk mengumpulkan brosur dan mendata ke *Microsoft Office* sebagai arsip data siswa baru. Oleh karena itu, dibuat sistem informasi PSB berbasis *website* dengan menggunakan *PHP*, *HTML*, dan *SQL*. Sistem ini akan membuat pelaksanaan PSB menjadi lebih transparan dan terhindar dari oknum-oknum yang melakukan kecurangan dan mengambil keuntungan dapat diminimalisir dengan sistem PSB online sehingga dapat mempermudah semua pihak agar tidak ada kecurangan dan kehilangan data siswa baru [7].

Penelitian sebelumnya juga pernah dilakukan di Raudhatul Athfal Sirojul Falah. Penelitian ini dilakukan karena sistem penerimaan siswa baru sekolah tersebut masih menggunakan metode konvensional. Petugas PPDB seringkali membutuhkan waktu lama untuk memasukkan data siswa ke dalam *Microsoft Excel* dan sering melakukan kesalahan pada program *Microsoft Excel* saat melapor kepada kepala sekolah. Pemrosesan dan pengolahan data masih membutuhkan waktu yang cukup lama, sehingga efisiensi pengolahan datanya belum tinggi. Untuk itu diperlukan suatu sistem informasi yang dapat membantu dalam proses PPDB. Dengan adanya PPDB berbasis sistem informasi maka data yang tersimpan akan jauh lebih aman, akurat serta menunjang segala kebutuhan. Proses mencari data siswa baru serta pengolahan data siswa bisa dilakukan lebih efektif dan efisien [8].

Penelitian yang dilakukan memiliki perbedaan dari penelitian-penelitian sebelumnya. Perbedaan tersebut diantaranya terdapat fungsi pendukung seperti penyampaian informasi jadwal PPDB, syarat dan ketentuan peserta didik baru, dan hasil pengumuman seleksi yang dapat dilihat di halaman *website*. Tujuan dari penelitian yang akan dilakukan yaitu membantu pihak sekolah dalam pendaftaran, penyampaian

informasi PPDB, dan penyampian hasil pengumuman seleksi. Selain itu membantu pihak orang tua dan calon peserta didik baru melakukan pendaftaran secara online. Sistem ini akan dikembangkan menggunakan bahasa pemrograman PHP dan *framework* Laravel, serta metode pengembangan *waterfall*, sistem yang akan dikembangkan menggunakan *MySQL* sebagai *database*, serta menggunakan metode pengujian sistem *black box*.

2.2 Landasan Teori

Penelitian yang dilakukan dilandaskan pada teori-teori yang berkaitan dengan inti dari penelitian. Berikut merupakan teori-teori yang digunakan sebagai acuan dalam penelitian :

2.2.1. Sistem Informasi

Sistem informasi adalah suatu cara tertentu untuk menyediakan informasi yang dibutuhkan oleh organisasi untuk beroperasi. Pada dasarnya sistem informasi merupakan suatu sistem yang dibuat oleh manusia yang terdiri dari komponen-komponen dalam organisasi untuk mencapai suatu tujuan yaitu menyajikan informasi. Sistem informasi di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan. Sistem informasi menerima masukan data dan instruksi, mengolah data tersebut sesuai instruksi, dan mengeluarkan hasilnya [9].

Sistem informasi adalah data yang dikumpulkan, dikelompokkan dan diolah sedemikian rupa sehingga menjadi sebuah satu kesatuan informasi yang saling terkait dan saling mendukung sehingga menjadi suatu informasi yang berharga bagi yang menerimanya. Pengertian lainnya, sistem informasi merupakan kumpulan elemen yang saling berhubungan satu sama lain yang membentuk satu kesatuan untuk mengintegrasikan data, memproses dan menyimpan serta mendistribusikan informasi [10]. Perkembangan sistem informasi meliputi sistem informasi tradisional yaitu sistem informasi yang dioperasikan dan dikelola secara semi-manual. Sistem informasi beroperasi secara lambat sehingga pengambilan keputusan sering berdasarkan asumsi atau perkiraan. Suatu sistem informasi pada dasarnya terbentuk oleh kelompok kegiatan operasi yang tetap, yaitu mengumpulkan data, mengelompokkan data, menghitung, menganalisa data, dan menyajikan laporan.

2.2.2. Rekayasa Perangkat Lunak

Tahap pengembangan sistem dalam aplikasi ini menggunakan metode *System Development Live Cycle* (SDLC) dengan model proses atau paradigma *Waterfall*. Metode *waterfall* dipilih karena melakukan pendekatan secara sistematis dan urut sehingga dalam melakukan pengerjaan proyek lebih terkontrol dan terjadwal dengan baik. Berikut adalah tahapan-tahapan dari metode *waterfall* [11]:

- a. *Analysis*
Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan Sistem Informasi Penerimaan Peserta Didik Baru berbasis *website* di SD Negeri Sidanegara 06 supaya dapat dipahami oleh *user*.
- b. *Design*
Design merupakan proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengkodean.
- c. *Coding*
Design harus ditranslasikan ke dalam program Sistem Informasi Penerimaan Peserta Didik Baru berbasis *website* di SD Negeri Sidanegara 06. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap *design*.
- d. *Testing*
Testing memfokuskan pada keperluan fungsional dari *software*, karena itu metode pengujian *black box* memungkinkan pengembangan *software*.
- e. *Maintenance*
Ketika Sistem Informasi Penerimaan Peserta Didik Baru berbasis *website* di SD Negeri Sidanegara 06 sudah dikirim ke *user*, perubahan dapat terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak tidak beradaptasi dengan lingkungan baru. Tahap *maintenance* dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada , tetapi tidak untuk membuat perangkat lunak baru.

Metode *black box* berfokus pada persyaratan fungsional perangkat lunak. *Black box testing* memungkinkan rekayasa perangkat lunak mendapatkan serangkaian kondisi *input* yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. *Black box testing*

merupakan pendekatan komplementer yang memungkinkan besar mampu mengungkap kelas kesalahan daripada metode *white box*. *Black box testing* berusaha menemukan kesalahan dalam kategori sebagai berikut [12] :

1. Fungsi-fungsi yang tidak benar atau hilang.
2. Kesalahan *interface*.
3. Kesalahan dalam struktur data atau akses *database* eksternal.
4. Kesalahan kinerja.
5. Inisialisasi dan kesalahan terminal.

Black box testing cenderung diaplikasikan selama tahap akhir pengujian. *Black box testing* memperhatikan struktur-struktur kontrol, maka perhatian berfokus pada domain informasi. *Black box testing* terdapat beberapa proses. Proses-proses tersebut diantaranya :

1. Menganalisa kebutuhan dan spesifikasi dari perangkat lunak.
2. Pemilihan jenis *input* yang memungkinkan menghasilkan output dengan benar serta jenis *input* yang memungkinkan *output* salah pada perangkat lunak yang sedang diuji.
3. Menentukan *output* untuk satu jenis *input*.
4. Pengujian dilakukan dengan *input-input* yang telah benar-benar diseleksi.
5. Melakukan pengujian.
6. Perbandingan *output* yang dihasilkan dengan *output* yang diharapkan.
7. Menentukan fungsionalitas yang seharusnya ada pada perangkat lunak yang sedang diuji.

Tahap desain atau perancangan sistem akan menggunakan *flowchart* dan desain berbasis UML. Berikut ini penjelasan *flowchart* dan desain berbasis UML :





A. Flowchart

Flowchart adalah bagan-bagan yang mempunyai arus yang menggambarkan langkah-langkah penyelesaian suatu masalah. *Flowchart* merupakan cara penyajian dari suatu algoritma. *Flowchart* disusun dengan simbol. Simbol ini dipakai sebagai alat bantu yang menggambarkan proses di dalam program. Simbol-simbol yang digunakan dapat dibagi menjadi 3 kelompok, yaitu [13]:

1. Flow Direction Symbols (Simbol Penghubung/Alur)

Simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga *connecting line*. Simbol-simbol tersebut adalah sebagai berikut :



Tabel 2. 1 Simbol Penghubung dalam *Flowchart*

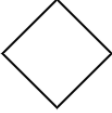
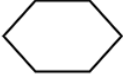




No.	Simbol	Keterangan
1.		Simbol arus / flow Untuk menyatakan jalannya arus suatu proses.
2.		Simbol Communication Link Untuk menyatakan bahwa adanya transisi suatu data atau informasi dari satu lokasi ke lokasi lainnya.
3.		Simbol Connector Untuk menyatakan sambungan dari satu proses ke proses lainnya dalam halaman atau lembar yang sama.
4.		Simbol Offline Connector Untuk menyatakan sambungan dari satu proses ke proses lainnya dalam halaman atau lembar yang berbeda.

2. *Processing Symbols (Simbol Proses)*

Simbol yang menunjukkan jenis operasi pengolahan dalam suatu proses atau prosedur. Simbol-simbol tersebut adalah sebagai berikut :

Tabel 2. 2 Simbol Proses dalam *Flowchart*



No.	Simbol	Keterangan
1.		Simbol Offline Connector Untuk menyatakan sambungan dari satu proses ke proses lainnya dalam halaman atau lembar yang berbeda.
2.		Simbol Manual Untuk menyatakan suatu tindakan (proses) yang tidak dilakukan oleh komputer (manual).





3.		Simbol <i>Decision</i> / Logika Untuk menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban, ya / tidak.
4.		Simbol <i>Predefined Proses</i> Untuk menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal.
5.		Simbol <i>Terminal</i> Untuk menyatakan permulaan atau akhir suatu program.
6.		Simbol <i>Keying Operation</i> Untuk menyatakan segala jenis operasi yang diproses dengan menggunakan suatu mesin yang mempunyai <i>keyboard</i> .
7.		Simbol <i>Off-line Storage</i> Untuk menunjukkan bahwa data dalam simbol ini akan disimpan ke dalam suatu media tertentu.
8.		Simbol <i>Manual Input</i> Untuk memasukkan data secara manual dengan menggunakan <i>online keyboard</i> .

3. *Input-output Symbols (Simbol Input-output)*

Simbol yang menunjukkan jenis peralatan yang digunakan sebagai media *input* atau *output*. Simbol-simbol tersebut adalah sebagai berikut :

Tabel 2. 3 Simbol Input-output dalam *Flowchart*

No.	Simbol	Keterangan
1.		Simbol <i>Input-output</i> Untuk menyatakan proses <i>input</i> dan <i>output</i> tanpa tergantung dengan jenis peralatannya.
2.		Simbol <i>Punched Card</i> Untuk menyatakan <i>input</i> berasal dari kartu atau <i>output</i> ditulis ke kartu.

3.		Simbol <i>Magnetic-tape Unit</i> Untuk menyatakan <i>input</i> berasal dari pita <i>magnetic</i> atau <i>output</i> disimpan ke pita <i>magnetic</i> .
4.		Simbol <i>Disk Storage</i> Untuk menyatakan <i>input</i> berasal dari <i>disk</i> atau <i>output</i> disimpan ke <i>disk</i> .
5.		Simbol <i>Document</i> Untuk mencetak laporan ke printer.
6.		Simbol <i>Display</i> Untuk menyatakan peralatan <i>output</i> yang digunakan berupa layar (video, komputer).

B. *Unified Modeling Language (UML)*








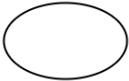
Unified Modeling Language (UML) adalah hasil kerja dari konsorium berbagai organisasi yang berhasil dijadikan sebagai standar baku dalam *Object Oriented Analysis* dan *Design (OOAD)*. Macam-macam UML antara lain [14]:



1. *Use Case Diagram*

Use case adalah deskripsi fungsi dari sebuah sistem dari perspektif pengguna. *Use case* bekerja dengan cara mendeskripsikan tipikal interaksi antar *user* (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sistem dipakai. Urutan langkah-langkah yang menerangkan antara pengguna dan sistem disebut *scenario* sedangkan pengguna disebut *actor*. *Actor* adalah sebuah peran yang biasa dimainkan oleh pengguna dalam interaksinya dengan sistem. *Use case* dibuat berdasarkan kebutuhan *actor*.

Berdasarkan definisi di atas dapat disimpulkan bahwa *use case* adalah kontruks untuk mendeskripsikan bagaimana sistem akan terlihat dimata pengguna potensial yang terdiri dari kumpulan *scenario* dan *actor*. *Use case diagram* memfasilitasi komunikasi diantara analis dan pengguna serta analis dan klien. Simbol-simbol *use case* antara lain :

Tabel 2. 4 Simbol *Use Case Diagram*





No.	Simbol	Keterangan
1.		Actor Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2.		Dependency Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung pada elemen yang tidak mandiri (<i>independent</i>).
3.		Generalization Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
4.		Include Menspesifikasikan <i>use case</i> sumber secara eksplisit.
5.		Extend Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6.		Association Menghubungkan antara objek satu dengan objek yang lainnya.
7.		System Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8.		Use Case Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i> .



9.		Collaboration Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10.		Note Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

2. *Sequence Diagram*

Sequence diagram digunakan untuk menggambarkan perilaku pada sebuah *scenario*. *Sequence diagram* menunjukkan sejumlah obyek dan *message* yang diletakkan antara obyek-obyek di dalam *use case*. Komponen *sequence diagram* terdiri dari objek yang dituliskan dengan kotak segiempat bernama. *Message* diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan *progress vertical*. *Sequence diagram* menambahkan dimensi waktu pada interaksi diantara obyek. Simbol-simbol yang dipakai dalam *sequence diagram* antara lain :

Tabel 2. 5 *Simbol Sequence Diagram*

No.	Simbol	Keterangan
1.		Actor Menggambarkan orang yang sedang bertinteraksi dengan sistem.
2.		Entity Class Menggambarkan hubungan yang akan dilakukan.
3.		Boundary Class Menggambarkan sebuah gambaran dari foem.
4.		Control Class Menggambarkan penghubung antara <i>boundary</i> dengan tabel.

5.		<p><i>A focus of Control & A Life Line</i> Menggambarkan tempat mulai dan berakhirnya <i>message</i>.</p>
6.		<p><i>A Message</i> Menggambarkan pengiriman pesan.</p>

2.2.3. Pemrograman Berorientasi Objek

Pemrograman berorientasi objek merupakan suatu metode pemrograman yang berbasis objek. Pada pemrograman berorientasi objek, antara objek yang satu berinteraksi dengan objek yang lain dengan tujuan mendapatkan informasi antarobjek. Pemrograman berorientasi objek adalah bagian dari pengembangan sistem berorientasi objek [15]. Tiga karakteristik utama dari metodologi pengembangan sistem berorientasi objek antara lain :

a. *Encapsulation*

Encapsulation (pengkapsulan) adalah dasar untuk pembatas ruang lingkup program terhadap data yang diproses. Data dan prosedur atau berfungsi dikemas bersama-sama dalam suatu objek, sehingga prosedur atau fungsi lain dari luar tidak dapat mengaksesnya. Data terlindung dari prosedur atau objek lain kecuali prosedur yang berada dalam objek itu sendiri.

b. *Inheritance*

Inheritance (pewarisan) merupakan teknik yang menyatakan bahwa anak dari objek akan mewariskan data atau atribut dan metode dari induknya langsung. Atribut dan metode dari objek induk diturunkan kepada anak objek, demikian seterusnya. Pendefinisian objek digunakan untuk membangun suatu hirarki dari objek turunannya, sehingga tidak diperlukan membuat atribut dan metode lagi pada anaknya, karena telah mewarisi sifat induknya.

c. *Polymorphism*

Polymorphism (polimorfisme) adalah konsep yang menyatakan bahwa sesuatu yang sama dapat mempunyai bentuk dan perilaku berbeda. Polimorfisme berarti bahwa operasi yang sama mungkin mempunyai perbedaan dalam kelas yang berbeda. Implementasi yang spesifik dari suatu operasi dari kelas tertentu disebut metoda.

Operator berorientasi objek bersifat polimorfisme yang mungkin dapat mempunyai lebih dari satu metoda.

2.2.4. Basis Data

Basis dapat diartikan sebagai markas atau gudang, tempat bersarang atau berkumpul. Sedangkan data merupakan representasi fakta dunia nyata yang mewakili suatu objek seperti manusia, barang, hewan, peristiwa, konsep, keadaan, dan sebagainya yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya. Basis data (*database*) merupakan kumpulan data yang saling berhubungan (punya relasi). Relasi biasanya ditunjukkan dengan kunci (*key*) dari setiap file yang ada. Basis data mempunyai beberapa kriteria penting yaitu, bersifat *data oriented* dan bukan *program oriented*, dapat digunakan oleh beberapa program aplikasi tanpa perlu mengubah basis datanya. Prinsip utama basis data adalah pengaturan data dengan tujuan utama fleksibilitas dan kecepatan dalam pengambilan data kembali [16].



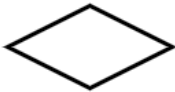

MySQL adalah sebuah program *database* server yang mampu menerima dan mengirim datanya dengan sangat cepat, multi *user* serta menggunakan perintah standar *Structured Query Language* (SQL). SQL memiliki 2 bentuk lisensi, yaitu *Free Software* dan *Shareware*. SQL yang biasa digunakan adalah *MySQL Free Software* yang berada di bawah Lisensi *General Public License* (GPL) [12].

MySQL dikembangkan oleh sebuah perusahaan Swedia bernama MySQL AB, yang pada saat itu bernama TcX Data Konsult AB tahun 1994-1995. MySQL sudah ada sejak tahun 1979. MySQL termasuk jenis *Relation Database Management System* (RDBMS) yang digunakan oleh banyak portal internet sebagai basis data dari informasi yang ditampilkan pada situs *web*. Kepopuleran MySQL dimungkinkan karena kemudahannya untuk digunakan, cepat secara kinerja *query*, dan mencukupi untuk kebutuhan basis data perusahaan-perusahaan skala menengah dan kecil.

A. *Entity Relationship Diagram* (ERD)

Entity Relationship Diagram (Diagram Hubungan antara Entitas) adalah suatu model jaringan yang menggunakan susunan data yang disimpan dalam sistem secara abstrak. ERD berbeda dengan DFD yang merupakan suatu model jaringan fungsi yang akan dilaksanakan oleh sistem, sedangkan ERD merupakan model jaringan data yang menekankan pada struktur-struktur dan relationship data [13]. Notasi-notasi simbolik di dalam diagram ERD yang dapat digunakan adalah sebagai berikut :

Tabel 2. 6 Simbol Entity Relationship Diagram

Simbol	Nama Simbol dan Fungsinya
	Himpunan Entitas adalah sesuatu apa saja yang ada di dalam sistem, nyata maupun abstrak dimana data tersimpan atau dimana terdapat data.
	Atribut adalah sifat atau karakteristik dari tiap entitas maupun tiap <i>relationship</i> .
	Himpunan Relasi adalah kumpulan semua relasi di antara entitas-entitas yang terdapat dalam himpunan entitas-himpunan entitas.
	Link adalah sebagai penghubung antara himpunan relasi dengan himpunan entitas dan himpunan entitas dengan atribut.

Elemen-elemen *Entity Relationship Diagram* (ERD) :

a. **Entity**

Entity digambarkan dengan sebuah bentuk persegi panjang. *Entity* adalah sesuatu apa saja yang ada di dalam sistem, nyata maupun abstrak dimana data tersimpan atau dimana terdapat data. Entitas diberi nama dengan kata benda dan dapat dikelompokkan dalam empat jenis nama, yaitu orang, benda, lokasi, kejadian (terdapat unsur waktu di dalamnya).

b. **Relationship**

Relationship dapat digambarkan dengan sebuah bentuk belah ketupat. *Relationship* adalah hubungan alamiah yang terjadi antara entitas. Pada umumnya, penghubung (*Relationship*) diberi nama dengan kata kerja dasar, sehingga memudahkan untuk melakukan pembacaan relasinya (bisa dengan kalimat aktif atau kalimat pasif). Penggambaran hubungan yang terjadi adalah sebuah bentuk belah ketupat dihubungkan dengan dua bentuk empat persegi panjang.

c. **Atribut**

Atribut adalah sifat atau karakteristik dari tiap entitas maupun tiap *relationship*. Maksudnya, atribut adalah sesuatu yang menjelaskan apa sebenarnya yang dimaksud entitas maupun

relationship, sehingga sering dikatakan atribut adalah elemen dari setiap entitas dan *relationship*.

d. Kardinalitas (*Cardinality*)

Kardinalitas relasi menunjukkan jumlah maksimum tupel yang dapat berelasi dengan entitas pada entitas yang lain. Dari sejumlah kemungkinan banyaknya hubungan antar entitas tersebut, Kardinalitas Relasi merujuk kepada hubungan maksimum yang terdiri dari entitas yang satu ke entitas yang lain dan begitu juga sebaliknya. Terdapat 3 macam kardinalitas relasi, yaitu :

1) *One to One*

Tingkat hubungan satu ke satu, dinyatakan dengan satu kejadian pada entitas pertama, hanya mempunyai satu hubungan dengan satu kejadian pada entitas yang kedua dan sebaliknya.

2) *One to Many* atau *Many to One*

Tingkat hubungan satu ke banyak adalah sama dengan banyak ke satu. Tergantung dari arah mana hubungan tersebut dilihat. Untuk satu kejadian pada entitas yang pertama dapat mempunyai banyak hubungan dengan kejadian pada entitas yang kedua. Sebaliknya satu kejadian pada entitas yang kedua hanya dapat mempunyai satu hubungan dengan satu kejadian pada entitas yang pertama.

3) *Many to Many*

Tingkat hubungan banyak ke banyak terjadi jika tiap kejadian pada sebuah entitas akan mempunyai banyak hubungan dengan kejadian pada entitas lainnya. Baik dilihat dari sisi entitas yang pertama, maupun dilihat dari sisi yang kedua.

2.2.5. Pendidikan Dasar

Pendidikan dasar sesuai dengan ketentuan UU No. 20 Tahun 2003 tentang Sistem Pendidikan Nasional Pasal 14, terdiri dari Sekolah Dasar (SD) atau Madrasah Ibtidaiyah (MI) dan Sekolah Menengah Pertama (SMP) atau Madrasah Tsanawiyah (MTs) atau bentuk lain yang sejenis. SD sebagai jenjang pendidikan dasar harus memiliki prinsip dan dasar yang kokoh karena akan menentukan kualitas pendidikan di jenjang pendidikan menengah. Prinsip dasar akan berjalan jika dibarengi dengan para pendidik dan tenaga kependidikan yang memahami hakikat prinsip dasar pendidikan dasar [17].

PPDB atau Penerimaan Peserta Didik Baru adalah metode pendaftaran sekolah dari tingkat PAUD, TK, SD, SMP, sampai SMA. Peraturan PPDB diterbitkan pemerintah melalui Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi (Kemendikbu) melalui Permendikbud Nomor 44 Tahun 2019 tentang Penerimaan Peserta Didik Baru pada Taman Kanak-Kanak, Sekolah Dasar, Sekolah Menengah Pertama, Sekolah Menengah Atas, dan Sekolah Menengah Kejuruan. Berdasarkan pasal 2, PPDB dilakukan berdasarkan nondiskriminatif, objektif, transparan, akuntabel, dan berkeadilan. Berdasarkan pasal 11, pendaftaran PPDB dilaksanakan melalui jalur zonasi, afirmasi, perpindahan tugas orang tua/wali, dan prestasi. Berdasarkan pasal 21, pelaksanaan PPDB dimulai dari tahap pengumuman pendaftaran penerimaan calon peserta didik baru dilakukan secara terbuka, pendaftaran, seleksi sesuai dengan jalur pendaftaran, pengumuman penetapan peserta didik baru, dan daftar ulang. Berdasarkan pasal 30, penetapan peserta didik baru dilakukan berdasarkan hasil rapat dewan guru yang dipimpin oleh kepala sekolah dan ditetapkan melalui keputusan kepala sekolah [18].