

BAB 2

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Penelitian sistem pakar untuk mendiagnosa suatu penyakit menggunakan metode *Dempster Shafer* yang telah dilakukan (Kurniawati, 2014) yang berjudul "Implementasi Metode *Dempster Shafer* Pada Sistem Pakar Untuk Diagnosa Jenis-jenis Penyakit Diabetes Melitus" menggunakan metode *Dempster Shafer* untuk mengidentifikasi jenis penyakit Diabetes Melitus. Menyimpulkan bahwa metode *Dempster Shafer* berhasil diterapkan untuk mendiagnosis berbagai jenis penyakit Diabetes Melitus berdasarkan gejala-gejala yang dialami pasien. Hasil uji coba menunjukkan bahwa sistem yang menggunakan metode ini menghasilkan diagnosis yang konsisten dengan hasil yang diberikan oleh pakar. Tingkat akurasi tertinggi yang dicapai oleh sistem adalah 96,67% dari 30 data yang diuji, mengindikasikan bahwa metode *Dempster Shafer* bekerja secara efektif dalam memberikan hasil diagnosis yang akurat dan sesuai dengan diagnosa pakar.

Penelitian sistem pakar mendiagnosa suatu penyakit menggunakan metode *Dempster Shafer* juga pernah dilakukan oleh (Budiman & Rismawan, 2016) yang berjudul "Aplikasi Sistem Pakar Untuk Mendiagnosa Penyakit Gangguan Saraf Dengan Metode *Dempster Shafer* Berbasis Android". Pada penelitian ini sistem memberikan informasi tentang penyakit apa saja yang terdapat pada gangguan saraf, hasil kesimpulan identifikasi dihitung menggunakan metode *Dempster Shafer*, sesuai dengan gejala-gejala yang telah dipilih oleh user, selanjutnya sistem akan menampilkan hasil kemungkinan penyakit beserta nilai kepercayaan atau densitas. Berdasarkan hasil pengujian dari dua rumah sakit yang berbeda maka dapat dihasilkan bahwa tingkat keberhasilan sistem secara umum adalah 90.9%.

Selain itu penelitian sistem pakar mendiagnosa penyakit demam berdarah dengue menggunakan metode yang berbeda dengan Metode *Dempster Shafer* pernah dilakukan oleh (Fuad et al., 2022) yang berjudul "Sistem Pakar Diagnosa

Penyakit Demam Berdarah Dengue Secara Dini Menggunakan Metode *Certainty Factor*”. Pada penelitian ini metode yang digunakan adalah metode *Certainty Factor*. Metode *Certainty Factor* digunakan ketika suatu masalah memiliki jawaban yang belum pasti atau masih mengandung ketidakpastian. Ketidakpastian ini sering disebut sebagai probabilitas, yang menggambarkan seberapa besar kemungkinan jawaban tersebut benar. Hasil kesimpulan pada penelitian ini menunjukkan bahwa sistem pakar untuk diagnosis dini penyakit demam berdarah dengue dengan metode *Certainty Factor* telah berhasil dibuat dengan baik. Metode *Certainty Factor* diterapkan dengan memberikan bobot pada setiap gejala yang ditentukan oleh pakar, sehingga menghasilkan persentase yang menunjukkan kemungkinan pasien terindikasi penyakit demam berdarah berdasarkan gejala-gejala yang muncul.

Penelitian sistem pakar menggunakan Metode *Certainty Factor* juga pernah dilakukan oleh (Arif et al., 2024) untuk mendiagnosa penyakit Chikungunya dengan judul “Diagnosa Penyakit Chikungunya Menggunakan Metode *Certainty Factor*”. Berdasarkan hasil penelitian mengenai sistem pakar untuk diagnosis penyakit Chikungunya menggunakan metode certainty factor adalah diagnosis dari gejala yang dipilih menunjukkan hasil paling akurat dengan persentase indikasi chikungunya akut sebesar 95,97%. Sistem pakar ini berhasil dikembangkan dan diterapkan dengan memanfaatkan pengetahuan dari para dokter dan dirancang sebagai alat bantu untuk mendeteksi gejala penyakit chikungunya sejak dini, sehingga memungkinkan adanya langkah pencegahan lebih awal.

Dari penelitian yang sudah ada hal ini menunjukkan bahwa diagnosis suatu penyakit menggunakan sistem pakar dapat memberikan hasil yang akurat sesuai dengan perhitungan pakar pada bidang kesehatan tersebut dan pastinya dapat bermanfaat bagi tenaga kesehatan untuk mendiagnosa suatu penyakit secara efisien.

Tabel 2.1 Hasil Penelitian Terdahulu

No.	Penulis	Judul	Persamaan dan perbedaan	Kesimpulan penelitian
1.	(Kurniawati, 2014)	Implementasi Metode <i>Dempster Shafer</i> Pada Sistem Pakar Untuk Diagnosa Jenis-jenis Penyakit Diabetes Melitus	<p>Persamaan: Menggunakan metode <i>Dempster Shafer</i> untuk analisis penyakit</p> <p>Perbedaan: Difokuskan pada identifikasi berbagai jenis penyakit Diabetes Melitus</p>	Penelitian ini menunjukkan bahwa metode <i>Dempster Shafer</i> berfungsi efektif dalam mengidentifikasi penyakit Diabetes Melitus, dengan akurasi hingga 96,67% berdasarkan 30 data uji. Sistem memberikan hasil yang konsisten dengan diagnosis pakar.
2.	(Budiman & Rismawan, 2016)	Aplikasi Sistem Pakar Untuk Mendiagnosa Penyakit Gangguan Saraf Dengan Metode <i>Dempster Shafer</i> Berbasis Android	<p>Persamaan: Menggunakan metode <i>Dempster Shafer</i> untuk analisis penyakit</p> <p>Perbedaan: Fokusnya pada diagnosa penyakit gangguan saraf</p>	Hasil penelitian dari dua rumah sakit menunjukkan keberhasilan sistem mendiagnosis penyakit gangguan saraf dengan tingkat akurasi 90,9%.
3.	(Fuad et al., 2022)	Sistem Pakar Diagnosa Penyakit Demam Berdarah Dengue Secara Dini Menggunakan	Persamaan: Mengidentifikasi salah satu jenis penyakit yang sama yaitu Demam	Sistem pakar menggunakan <i>Certainty Factor</i> untuk diagnosis dini penyakit Demam Berdarah Dengue berhasil

		Metode <i>Certainty Factor</i>	Berdarah Dengue	diterapkan dengan baik.
			Perbedaan: Menggunakan metode <i>Certainty Factor</i> analisis penyakit	Sistem pakar menggunakan <i>Certainty Factor</i> untuk diagnosis dini penyakit Demam Berdarah Dengue berhasil diterapkan dengan baik. Sistem ini memberikan hasil akurat tentang kemungkinan penyakit berdasarkan gejala.
4.	(Arif et al., 2024)	Diagnosa Penyakit Chikungunya Menggunakan Metode <i>Certainty Factor</i>	Persamaan: Mengidentifikasi salah satu jenis penyakit yang sama yaitu Chikungunya Perbedaan: Menggunakan metode <i>Certainty Factor</i> analisis penyakit	Penelitian ini menyimpulkan bahwa metode <i>Certainty Factor</i> efektif untuk diagnosis Chikungunya, dengan tingkat indikasi hingga 95,97%. Sistem ini dirancang untuk membantu diagnosis dini yang dapat mendorong pencegahan lebih awal.

2.2 Sistem Pakar

2.2.1 Pengertian Sistem Pakar

Sistem pakar adalah salah satu bagian dari kecerdasan buatan (Artificial Intelligence) yang bertujuan untuk mengintegrasikan pengetahuan manusia ke dalam komputer, sehingga komputer mampu memecahkan masalah dengan cara

yang mirip seperti seorang ahli. Sistem ini dirancang dan diimplementasikan menggunakan bahasa pemrograman tertentu untuk memungkinkan komputer menyelesaikan masalah layaknya seorang pakar yang memiliki pengetahuan dan keahlian khusus dalam bidang tertentu (Arhami, 2005). Sistem pakar dirancang untuk memberikan saran dan membantu menyelesaikan masalah di berbagai bidang spesialisasi, seperti sains, teknik, matematika, kedokteran, dan pendidikan.

2.2.2 Ciri-ciri Sistem Pakar

Sistem pakar yang berkualitas idealnya memiliki beberapa karakteristik utama, yaitu (Fadli, 2010) :

1. Menyediakan informasi yang akurat dan dapat diandalkan, baik dalam menyajikan langkah-langkah penyelesaian maupun dalam menjawab pertanyaan terkait proses tersebut.
2. Mampu menerapkan prinsip heuristik, sehingga bisa memanfaatkan pengetahuan untuk menemukan solusi secara efektif.
3. Memiliki fleksibilitas untuk digunakan pada berbagai jenis perangkat komputer.
4. Mudah diubah atau diperbarui, termasuk dalam hal penambahan atau penghapusan pengetahuan dari basis datanya.
5. Beradaptasi terhadap situasi yang berubah, memberikan kemampuan untuk tetap relevan sesuai perkembangan.

2.2.3 Kelebihan Sistem Pakar

Sistem pakar dapat memberikan beberapa keuntungan diberbagai bidang, diantaranya (Hidayati, 2010):

1. Memungkinkan orang awam untuk dapat bekerja seperti layaknya seorang pakar.
2. Mampu bekerja dengan informasi yang tidak lengkap atau tidak pasti.

3. Memberikan saran yang konsisten dan mengurangi kemungkinan terjadinya kesalahan.
4. Membuat pengoperasian alat yang kompleks menjadi lebih mudah, serta dapat melatih pekerja yang kurang berpengalaman.
5. Mengurangi waktu yang dibutuhkan dalam proses pengambilan keputusan.

2.2.4 Kekurangan Sistem Pakar

Seperti halnya sistem lain, sistem pakar juga memiliki beberapa kelemahan (Sulistiyohati & Hidayat, 2008):

1. Kesulitan dalam memperoleh pengetahuan, karena tidak selalu mudah untuk mendapatkan informasi, apalagi jika ahli di bidang terkait tidak tersedia atau metode yang digunakan oleh ahli berbeda-beda.
2. Membangun sistem pakar berkualitas tinggi sangat menantang dan membutuhkan biaya besar untuk pengembangan dan pemeliharannya.
3. Meskipun bermanfaat, sistem pakar tidak sepenuhnya sempurna, dan manusia tetap memiliki peran penting karena tidak semua keputusan yang diambil sistem selalu benar.

2.2.5 Pengertian Diagnosa

Diagnosa adalah proses identifikasi jenis penyakit yang dilakukan dengan memeriksa gejala-gejala yang muncul. Dari sisi medis, diagnosis mencakup proses penentuan kondisi atau penyakit berdasarkan tanda dan gejala pasien, menggunakan metode serta alat diagnostik seperti tes laboratorium, pencitraan, dan pemeriksaan klinis (KBBI, n.d.).

2.3 Dempster Shafer

Metode ini bekerja dengan menggabungkan berbagai bukti atau gejala yang diberikan oleh pakar dan mengubahnya menjadi nilai kepercayaan (*belief*) tertentu untuk memperoleh hasil yang lebih presisi. Pendekatan ini sangat bermanfaat dalam

sistem pakar, terutama ketika informasi yang tersedia tidak lengkap atau ambigu, seperti pada penyakit dengan gejala yang serupa, misalnya Demam Berdarah Dengue (DBD) dan Chikungunya (Surya Dhoni et al., n.d.).

Selain itu, metode *Dempster Shafer* dikenal karena kemampuannya untuk mengintegrasikan bukti yang mungkin saling bertentangan, kemudian menghitung tingkat kemungkinan (*plausibility*) dari diagnosis. Ini membuat metode tersebut sangat efektif dalam mengurangi ketidakpastian selama proses pengambilan keputusan, sehingga menjadi pilihan yang tepat untuk digunakan dalam sistem pakar medis (Surya Dhoni et al., n.d.).

Secara garis besar, teori *Dempster Shafer* disusun dalam interval *Belief* dan *Plausibility*. *Belief* (Bel) mencerminkan tingkat keyakinan atau dukungan dari *evidence* terhadap suatu kelompok proposisi tertentu. Ketika *Belief* bernilai 0, hal ini menunjukkan tidak adanya *evidence* yang mendukung, sedangkan nilai 1 mengindikasikan adanya kepastian penuh. Sebaliknya, *Plausibility* (Pls) berfungsi mengurangi tingkat kepastian dari *evidence*, sehingga menunjukkan potensi keyakinan yang lebih rendah atas proposisi yang diberikan. (Sinaga & Sembiring, 2016). Fungsi *Belief* dapat diformulakan sebagai berikut (Giarratano & Riley, 2005):

$$Bel(X) = \sum_{Y \subseteq X} m(Y)$$

Keterangan:

Bel(X) : *Belief* (X)

m(Y) : m(Y) = *mass function* dari (Y)

Sedangkan *Plausibility* (Pls) diformulakan sebagai berikut :

$$Pls(X) = 1 - Bel(X) = 1 - \sum_{Y \not\subseteq X} m(Y)$$

Bel(X') : *Belief* (X)

Pls(X) : *Plausibility* (X)

m(X') : *mass function* dari (X)

m(Y) : *mass function* dari (Y)

Nilai *plausibility* berkisar antara 0 hingga 1. Jika terdapat keyakinan penuh terhadap X , maka dapat dinyatakan bahwa $belief(X) = 1$, sehingga dari persamaan tersebut, nilai $pls(X) = 0$. Beberapa contoh rentang antara *belief* dan *plausibility* disajikan pada Tabel 2.2 (Giarratano & Riley, 2005).

Tabel 2.2 Range *Belief* dan *Plausability*

Kemungkinan	Keterangan
[1,1]	Semua benar
[0,0]	Semua benar
[0,1]	Ketidakpastian
Bel, 1] where $0 < Bel < 1$	Cenderung mendukung
[0, Pls] where $0 < Pls < 1$	Cenderung menolak
[Bel, Pls] where $0 < Bel \leq Pls < 1$	Cenderung mendukung dan menolak

Jika terdapat lebih dari satu fungsi *densitas*, maka digunakan rumus kombinasi. Misalnya, jika X merupakan bagian dari Θ dengan m_1 sebagai fungsi *densitas*, dan Y merupakan bagian dari Θ dengan m_2 sebagai fungsi *densitas*, maka fungsi kombinasi antara m_1 dan m_2 dapat dibentuk menjadi m_3 sebagai berikut (Giarratano & Riley, 2005):

$$m_3(Z) = \frac{\sum_{X \cap Y = Z} m_1(X) \cdot m_2(Y)}{1 - \sum_{X \cap Y = P} m_1(X) \cdot m_2(Y)}$$

Dimana:

$m_3(Z)$ = mass function dari evidence (Z)

$m_1(X)$ = mass function dari evidence (X), yang diperoleh dari nilai keyakinan suatu evidence dikalikan dengan nilai *disbelief* dari evidence tersebut.

$m_2(Y)$ = mass function dari evidence (Y), yang diperoleh dari nilai keyakinan suatu evidence dikalikan dengan nilai *disbelief* dari evidence tersebut.

$\sum_{X \cap Y = Z} m_1(X) \cdot m_2(Y)$ = merupakan nilai kekuatan dari evidence Z yang diperoleh dari hasil kombinasi nilai keyakinan sekumpulan evidence.

Langkah-langkah penyelesaian Metode Dempster Shafer antara lain (Giarratano & Riley, 2005):

- 1) Tentukan nilai *belief* gejala pada setiap penyakit dengan rumus :

$$Bel(X) = \sum_{Y \subseteq X} m(Y)$$

- 2) Hitung nilai *plausability*

$$Pls(X) = 1 - Bel(X) = 1 - \sum_{Y \not\subseteq X} m(Y)$$

- 3) Hitung nilai kombinasi *mass function* m1 dan m2 sebagai m3

$$m_3(Z) = \frac{\sum_{X \cap Y = Z} m_1(X) \cdot m_2(Y)}{1 - \sum_{X \cap Y = \emptyset} m_1(X) \cdot m_2(Y)}$$

- 4) Hasil diagnosa penyakit

Beberapa penelitian sebelumnya menyoroti keunggulan metode *Dempster Shafer* dalam menangani ketidakpastian dan ketidakkonsistenan yang sering muncul dalam proses diagnosis penyakit. Banyak penelitian memilih metode ini karena selain tingkat akurasi yang tinggi, *Dempster Shafer* juga menawarkan fleksibilitas dalam menangani perubahan data atau munculnya gejala baru dalam proses diagnosis.

2.4 Analisis Penyakit Akibat Gigitan Nyamuk

Dalam mendiagnosis suatu penyakit, langkah awal yang biasa dilakukan adalah mengenali gejala-gejala umum yang sering muncul pada penyakit tertentu. Meskipun hanya mengandalkan indikasi dari gejala klinis, seorang dokter umumnya mampu menentukan jenis penyakit yang diderita oleh pasien. Namun, pada beberapa kasus tertentu, pemeriksaan tambahan tetap dibutuhkan untuk memastikan hasil diagnosis secara lebih akurat (Novita Joseph, 2024). Berikut ini jenis penyakit yang ditularkan melalui gigitan nyamuk disertai dengan gejala yang menyertainya dapat dilihat sebagai berikut:

1. Demam Berdarah Dengue (DBD)

Demam Berdarah Dengue disebabkan oleh virus dengue yang ditularkan melalui gigitan nyamuk *Aedes aegypti*. Penularan terjadi saat nyamuk yang terinfeksi menggigit manusia dan menyebarkan virus ke dalam tubuh (Novita Joseph, 2024).

Gejala yang umum muncul antara lain:

- Demam tinggi ≥ 38 derajat (2-7hr)
- Sakit Kepala
- Nyeri belakang mata
- Badan lemas
- Kulit ruam (kemerah-merahan)

Pencegahan terbaik terhadap DBD adalah dengan menghindari gigitan nyamuk serta melakukan upaya pengendalian populasi nyamuk di lingkungan sekitar (Novita Joseph, 2024).

2. Chikungunya

Chikungunya adalah penyakit akibat virus chikungunya yang disebarkan oleh nyamuk *Aedes aegypti* dan *Aedes albopictus*. Virus ini menyebar melalui gigitan nyamuk yang telah terinfeksi (Novita Joseph, 2024).

Gejala yang umum muncul antara lain:

- Demam tinggi ≥ 38 derajat (2-7hr)
- Nyeri sendi hebat
- Sakit kepala
- Nyeri otot
- Mual dan muntah

Tanda-tanda infeksi biasanya muncul dalam 2 hingga 7 hari setelah seseorang digigit oleh nyamuk pembawa virus. Gejalanya bisa sangat mengganggu

aktivitas sehari-hari karena menimbulkan rasa sakit yang hebat pada persendian (Novita Joseph, 2024).

3. Malaria

Malaria merupakan penyakit serius yang disebabkan oleh parasit *Plasmodium*, yang ditularkan melalui gigitan nyamuk *Anopheles* betina. Saat nyamuk tersebut menggigit, parasit masuk ke dalam aliran darah dan menginfeksi sel darah merah (Novita Joseph, 2024).

Gejala yang umum muncul antara lain:

- Demam tinggi ≥ 38 derajat (2-7hr)
- Sakit kepala
- Menggigil atau kedinginan hebat
- Mual dan muntah
- Sakit Perut

Malaria adalah penyakit berbahaya yang umumnya ditemukan di wilayah tropis dan subtropis. Penularannya dapat berakibat fatal apabila tidak ditangani secara cepat dan tepat (Novita Joseph, 2024).

4. Filariasis (Kaki Gajah)

Filariasis merupakan penyakit menahun yang disebabkan oleh infeksi cacing filaria seperti *Brugia timori*, *Brugia malayi*, dan *Wuchereria bancrofti*. Penularan terjadi melalui gigitan nyamuk dari genus *Culex*, *Anopheles*, *Mansonia*, dan *Aedes* yang membawa larva cacing (Novita Joseph, 2024).

Gejala yang umum muncul antara lain:

- Demam tinggi ≥ 38 derajat (2-7hr)
- Radang saluran kelenjar
- Kelenjar Bengkak
- Absesfilarial

- Pembengkakan dini

Penyakit ini berkembang secara bertahap dan dapat menyebabkan pembengkakan kronis, bahkan mempengaruhi kualitas hidup serta fungsi tubuh secara keseluruhan jika tidak segera ditangani (Novita Joseph, 2024).

2.5 HTML (*HYPERTEXT MARKUP LANGUAGE*)

HTML, atau Hyper Text Markup Language, adalah bahasa markup yang digunakan untuk merancang halaman web dan menampilkan berbagai jenis informasi di dalam peramban internet. HTML berawal dari bahasa markup yang telah banyak digunakan di dunia penerbitan dan percetakan, yaitu SGML (Standard Generalized Markup Language). Sebagai standar yang digunakan luas untuk membangun halaman web, HTML saat ini dikelola dan ditetapkan sebagai standar internet oleh World Wide Web Consortium (W3C) (Rio Jumardi, n.d.).

HTML terdiri dari kode-kode tag yang memberikan instruksi kepada peramban untuk menampilkan halaman sesuai dengan tata letak yang diinginkan. File HTML dapat dibuka menggunakan berbagai browser web seperti Mozilla Firefox atau Microsoft Internet Explorer. Selain itu, HTML juga dikenali oleh aplikasi pembaca email, perangkat PDA, serta berbagai program lain yang memiliki kemampuan untuk menampilkan halaman web (Rio Jumardi, n.d.).

2.6 MYSQL

MySQL adalah sistem manajemen basis data relasional (RDBMS) yang bersifat open-source. Proyek ini dikembangkan pada tahun 1995 oleh Michael “Monty” Widenius, David Axmark, dan Allan Larson, yang kemudian mendirikan perusahaan MySQL AB di Swedia untuk mengelola dan mempromosikan perangkat lunak ini (Saputra et al., n.d.).

Pengertian MySQL menurut MySQL manual adalah perangkat lunak database open-source berbasis SQL (Structured Query Language) yang mengelola

sistem basis data relasional. Diciptakan oleh perusahaan asal Swedia, MySQL AB, perangkat ini dirancang untuk mudah dipahami oleh pengguna serta untuk menangani basis data besar dengan cepat dan efisien. Dengan kecepatan, konektivitas, serta keamanan yang tinggi, MySQL menjadi pilihan populer untuk mengakses basis data di internet (Reese, 2003).

Versi pertama MySQL, yaitu versi 1.0, diluncurkan pada Mei 1996, tetapi penggunaannya awalnya terbatas untuk perusahaan tertentu saja. Kemudian pada Oktober 1996, MySQL versi 3.11.0 dirilis untuk publik. MySQL menggunakan SQL (Structured Query Language) sebagai bahasa standar dalam mengelola data secara interaktif. Dengan performa, kecepatan, dan keandalannya, MySQL bersaing dengan database besar lainnya seperti ORACLE, Sybase, dan Unify. Selain itu, MySQL dapat dijalankan di berbagai sistem operasi, termasuk Linux, Windows, Solaris, FreeBSD, dan Mac OS X (Reese, 2003).

2.7 PHP

PHP merupakan salah satu bahasa pemrograman skrip yang paling banyak digunakan, terutama dalam pengembangan situs web, meskipun dapat juga diterapkan pada keperluan lain. PHP termasuk dalam kategori server-side scripting yang disisipkan di antara HTML untuk membangun halaman web yang dinamis. Karena berbasis server-side, PHP dijalankan di sisi server, sehingga yang ditampilkan ke pengguna adalah hasil HTML yang dihasilkan, tanpa menampilkan kode PHP. PHP berfungsi, antara lain, untuk menerima, memproses, dan menampilkan data di situs web. Data yang diterima akan diolah di server database, kemudian hasilnya akan muncul di layar peramban web (Arief, 2011).

Selain PHP, tersedia alternatif teknologi serupa yang memiliki keunggulan dan kelemahan masing-masing, antara lain (Arief, 2011):

1. ASP (Active Server Pages)
2. CFML (Cold Fusion Markup Language)

3. JSP (Java Server Pages)

2.8 LARAVEL

Laravel merupakan salah satu *framework* pengembangan aplikasi web berbasis arsitektur MVC yang dibuat menggunakan bahasa pemrograman PHP. Tujuan dari Laravel adalah untuk membantu pengembang menciptakan perangkat lunak yang berkualitas, dengan cara menekan biaya pengembangan awal maupun biaya pemeliharaan di kemudian hari. Selain itu, Laravel menawarkan sintaks yang mudah dipahami dan fungsionalitas inti yang lengkap, sehingga proses pengembangan bisa dilakukan lebih cepat dan efisien (McCool, 2012).

Salah satu keunggulan Laravel dibandingkan kerangka kerja PHP lainnya adalah sifatnya yang *modular*. Hal ini dimungkinkan berkat sistem *driver* dan *bundle* yang dimilikinya. *Driver* mempermudah pengembang dalam melakukan kustomisasi dan pengembangan fitur seperti *caching*, *session*, *database*, serta sistem autentikasi. Sementara itu, penggunaan *bundle* memungkinkan pengemasan berbagai kode agar bisa digunakan ulang atau dibagikan kepada komunitas pengguna Laravel. Yang membuat Laravel semakin fleksibel adalah kemampuan untuk mengemas semua komponen yang dibuat ke dalam satu paket yang bisa dipakai kembali (McCool, 2012).