

LAMPIRAN

1. *Code* untuk pengambilan data kriteria 1 sampai dengan kriteria 4.

Pengambilan data kriteria dalam *database* mengacu pada table database BTS dengan relasi setiap kriteria nya. Dalam kriteria 1 yaitu pingtime, relasi dari *database* adalah *table pops* dan pings dimana data pingtime yang diambil adalah *average* dari 1 bulan pings dan pings simultan dilakukan oleh sistem dengan waktu 10 menit sekali. Kriteria 2 yaitu total *accesspoint*, relasi dari *database* adalah *table pops* dan *accesspoints* dimana data semua *accesspoint* dari setiap BTS di totalkan. Kriteria 3 yaitu total *customer*, relasi dari *database* adalah *table pops* dan *customers* dimana data semua customer dari setiap BTS di totalkan. Dan yang terakhir kriteria 4 yaitu total ticket, relasi dari *database* adalah *table pops* dan *tickets*, dimana data ticket yang diambil adalah total dari 1 bulan ticket berdasarkan dari relasi BTS tersebut. *Source code* dapat dilihat di Lampiran 1.

```

$criteria1 = DB::select(
    DB::raw(
        "SELECT pops.id, pops.nama, AVG(a.pingtime) as avg_pingtime
        FROM (SELECT * FROM
            pings
            WHERE month(`created_at`) = ".date('m')." AND year(`created_at`)
            = ".date('Y')." ) a
        RIGHT JOIN `pops` ON `pops`.`ipAddr` = a.`ipAddr`
        GROUP BY `pops`.`id`
        ORDER BY `pops`.`id` asc"
    ));
$criteria2 = DB::select(
    DB::raw(
        "SELECT pops.nama, COUNT(a.pop_id) as total_ap
        FROM (SELECT * FROM
            accesspoints) a
        RIGHT JOIN `pops` ON `pops`.`id` = a.`pop_id`
        GROUP BY `pops`.`id`
        ORDER BY `pops`.`id` asc"
    ));
$criteria3 = DB::select(
    DB::raw(
        "SELECT pops.nama, COUNT(a.pop_id) as total_client
        FROM (SELECT * FROM
            customers) a
        RIGHT join `pops` ON `pops`.`id` = a.`pop_id`
        GROUP BY `pops`.`id`
        ORDER BY `pops`.`id` asc"
    ));
$criteria4 = DB::select(
    DB::raw(
        "SELECT pops.nama, COUNT(a.pop_id) as total_ticket
        FROM (SELECT * FROM
            tickets
            WHERE month(`start_downtime`) = ".date('m')." AND
            year(`start_downtime`) = ".date('Y')." ) a
        RIGHT join `pops` ON `pops`.`id` = a.`pop_id`
        GROUP BY `pops`.`id`
        ORDER BY `pops`.`id` asc"
    ));

```

Lampiran 1 Source Code Pengambilan Data Kriteria

2. Code untuk pemisahan nilai bobot kriteria.

Setelah mendapatkan data setiap kriteria, selanjutnya memisahkan data tersebut berdasarkan bobot yang telah ditentukan pada Tabel 4.1, Tabel 4.2, Tabel 4.3 dan Tabel 4.4. Source code dapat dilihat pada Lampiran 2.

```
private function getCriteriaValue($criteria, $value)
{
    switch ($criteria) {
        case 1:
            if ($value <= 30) {
                return 0;
            } elseif ($value > 30 && $value <= 70) {
                return 0.25;
            } elseif ($value > 70 && $value <= 150) {
                return 0.5;
            } elseif ($value > 150 && $value <= 250) {
                return 0.75;
            } else {
                return 1;
            }
            break;

        case 2:
            if ($value <= 5) {
                return 0;
            } elseif ($value > 5 && $value <= 15) {
                return 0.25;
            } elseif ($value > 15 && $value <= 25) {
                return 0.5;
            } elseif ($value > 25 && $value <= 50) {
                return 0.75;
            } else {
                return 1;
            }
            break;

        case 3:
            if ($value <= 25) {
                return 0;
            } elseif ($value > 25 && $value <= 70) {
                return 0.25;
            } elseif ($value > 70 && $value <= 125) {
```

```

        return 0.75;
    } else {
        return 1;
    }
    break;

case 4:
    if ($value <= 25) {
        return 0;
    } elseif ($value > 25 && $value <= 75) {
        return 0.25;
    } elseif ($value > 75 && $value <= 125) {
        return 0.5;
    } elseif ($value > 125 && $value <= 300) {
        return 0.75;
    } else {
        return 1;
    }
    break;
}
}

```

Lampiran 2 Source Code Pemisahan Nilai Bobot Kriteria

3. *Code untuk perhitungan Simple Additive Weighting.*

Setelah mendapatkan data setiap kriteria dan memisahkan nilai bobot sesuai dengan ketentuan yang telah dibuat, penghitungan Simple Additive Weighting dapat dilihat pada Lampiran 3.

```

$data = array();
$pop_max_total = "";
$max_total = 0;
foreach ($criteria1 as $key => $value) {
    $scr1 = $this->getCriteriaValue(1, $criteria1[$key]->avg_pingtime)*0.15;
    $scr2 = $this->getCriteriaValue(2, $criteria2[$key]->total_ap)*0.32;
    $scr3 = $this->getCriteriaValue(3, $criteria3[$key]->total_client)*0.3;
    $scr4 = $this->getCriteriaValue(4, $criteria4[$key]->total_ticket)*0.23;
    $total = $scr1+$scr2+$scr3+$scr4;

    if ($total > $max_total) {
        $max_total = $total;
        $pop_max_total = $criteria1[$key]->nama;
    }

    $data['pops'][] = [
        'id' => $criteria1[$key]->id,
        'nama' => $criteria1[$key]->nama,
        'criteria1' => $scr1,
        'criteria2' => $scr2,
        'criteria3' => $scr3,
        'criteria4' => $scr4,
        'total' => $total
    ];
}

```

Lampiran 3 Source Code Simple Additive Weighting

Halaman Ini Sengaja Dikosongkan